[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 15/17 (2006.01)
G06F 13/28 (2006.01)
G06F 12/00 (2006.01)

[12] 发明专利申请公布说明书

[21] 申请号 200710165058.1

[43] 公开日 2008 年 6 月 25 日

[11] 公开号 CN 101206633A

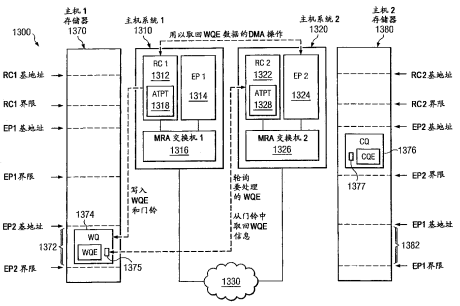
[22] 申请日 2007.11.6
[21] 申请号 200710165058.1
[30] 优先权
[32] 2006.12.19 [33] US [31] 11/612,530
[71] 申请人 国际商业机器公司
地址 美国纽约阿芒克
[72] 发明人 D·M·弗赖穆特 R·J·雷西奥
C·A·萨尔兹伯格 S·M·瑟伯
J·A·瓦尔加斯
[74] 专利代理机构 北京市金杜律师事务所
代理人 王茂华
权利要求书 4 页 说明书 49 页 附图 25 页

[54] 发明名称

用事务协议和共享存储器在主机系统间通信的系统和方法

[57] 摘要

提供了一种用于使用事务协议和共享存储器在主机系统之间进行通信的系统和方法。 在通信架构中基于发现过程来初始化共享存储器，使得至少一个端点在至少两个主机系统的共享存储器中具有地址范围。 可以建立面向事务的协议，用以使用主机系统的共享存储器来在相同或不同主机系统的根联合体和端点之间进行通信。 面向事务的协议指定了由例如根联合体或端点的各种元件所执行的一系列事务，用以推送或拉回数据。 可以使用推送和拉回事务的各种组合。



1. 一种在数据处理系统中用于在第一主机系统和第二主机系统之间通信的方法，该方法包括：

初始化在第一共享存储器中的第一存储器地址空间，以包括分配给与所述第二主机系统相关联的端点的第一地址范围；

初始化在第二共享存储器中的第二存储器地址空间，以包括分配给与所述第二主机系统相关联的端点的第二地址范围；

在所述第一地址范围中生成工作队列结构；

在所述第二地址范围中生成完成队列结构，所述完成队列结构包括第二门铃结构；以及

由所述第一主机系统和第二主机系统根据建立的事务协议来执行推送或拉回操作中的至少一个，以在所述工作队列结构和所述第二主机系统之间以及在所述完成队列结构和所述第一主机系统之间传送工作队列单元和完成队列单元。

2. 根据权利要求 1 所述的方法，其中所述建立的事务协议是拉回-拉回-推送事务协议。

3. 根据权利要求 2 所述的方法，其中根据所述拉回-拉回-推送事务协议：

所述端点执行直接存储器访问操作，以将工作队列单元从所述工作队列结构拉回到所述端点；

所述端点执行直接存储器访问操作，以将与工作队列单元对应的数据从所述第一共享存储器拉回；以及

所述端点执行操作，以将完成队列单元推送到所述第一主机系统。

4. 根据权利要求 1 所述的方法，其中所述建立的事务协议由所述端点执行。

5. 根据权利要求 1 所述的方法，其中所述第一地址范围和第二地址范围是可由主机系统经由存储器映射的输入/输出（I/O）操作和

至少一个地址转换和保护表来访问的。

6. 根据权利要求 1 所述的方法，其中所述推送操作和拉回操作是直接存储器访问（DMA）操作。

7. 根据权利要求 1 所述的方法，其中初始化与第一主机系统相关联的第一存储器地址空间和初始化与第二主机系统相关联的第二存储器地址空间包括：

遍历所述数据处理系统的通信架构中的链路，以收集关于在通信架构中存在的端点和根联合体的信息；

生成至少一个虚拟层级，所述虚拟层级标识在物理上或逻辑上相互关联的至少一个端点和至少一个根联合体；

基于所述至少一个虚拟层级来初始化所述第一存储器地址空间和所述第二存储器地址空间，使得与所述第一主机系统的根联合体相关联的每个端点在所述第一存储器地址空间中具有相应的地址范围，以及与所述第二主机系统的根联合体相关联的每个端点在所述第二存储器地址空间中具有相应的地址范围。

8. 根据权利要求 7 所述的方法，其中所述至少一个端点在所述第一存储器地址空间和所述第二存储器地址空间两者中都具有相应的地址范围。

9. 根据权利要求 1 所述的方法，其中所述数据处理系统是刀片服务器，以及所述第一主机系统和所述第二主机系统是所述刀片服务器中的刀片。

10. 根据权利要求 1 所述的方法，其中所述数据处理系统包括所述第一主机系统和所述第二主机系统与之耦合的外围组件互连 Express（PCIe）架构，并且其中所述端点是 PCIe 适配器。

11. 根据权利要求 10 所述的方法，其中所述 PCIe 架构是包括一个或多个多根感知（MRA）交换机的多根感知 PCIe 架构。

12. 一种计算机程序产品，包括具有计算机可读程序的计算机可用介质，其中当在所述数据处理系统中执行所述计算机可读程序时，将使所述数据处理系统执行根据权利要求 1-11 任何一个的方法步

骤。

13. 一种数据处理系统，包括：

第一主机系统；

第二主机系统；以及

耦合到所述第一主机系统和所述第二主机系统的通信架构，其中：

在第一共享存储器中初始化第一存储器地址空间，以包括分配给与所述第二主机系统相关联的端点的第一地址范围；

在第二共享存储器中初始化第二存储器地址空间，以包括分配给与所述第二主机系统相关联的端点的第二地址范围；

在所述第一地址范围中生成工作队列结构；

在所述第二地址范围中生成完成队列结构，所述完成队列结构包括第二门铃结构；以及

所述第一主机系统和所述第二主机系统根据建立的事务协议来执行推送或拉回操作中的至少一个，以在所述工作队列结构和所述第二主机系统之间以及在所述完成队列结构和所述第一主机系统之间来传送工作队列单元和完成队列单元。

14. 根据权利要求 13 所述的数据处理系统，其中所述建立的事务协议是拉回-拉回-推送事务协议。

15. 根据权利要求 14 所述的数据处理系统，其中根据所述拉回-拉回-推送事务协议，所述端点：

执行直接存储器访问操作，以将工作队列单元从所述工作队列结构拉回到所述端点；

执行直接存储器访问操作，以将与所述工作队列单元对应的数据从所述第一共享存储器拉回；以及

执行操作以将完成队列单元推送到所述第一主机系统。

16. 根据权利要求 13 所述的数据处理系统，其中所述建立的事务协议由所述端点执行。

17. 根据权利要求 13 所述的数据处理系统，其中所述第一地址

范围和第二地址范围是可由主机系统经由存储器映射的输入/输出（I/O）操作以及至少一个地址转换和保护表来访问的。

18. 根据权利要求 13 所述的数据处理系统，其中所述推送操作和拉回操作是直接存储器访问（DMA）操作。

19. 根据权利要求 13 所述的数据处理系统，其中通过如下动作来初始化与所述第一主机系统相关联的所述第一存储器地址空间和与所述第二主机系统相关联的所述第二存储器地址空间：

遍历所述数据处理系统的通信架构中的链路，以收集关于所述通信架构中存在的端点和根联合体的信息；

生成至少一个虚拟层级，所述虚拟层级标识在物理上或逻辑上相互关联的至少一个端点和至少一个根联合体；以及

基于所述至少一个虚拟层级来初始化所述第一存储器地址空间和所述第二存储器地址空间，使得与所述第一主机系统的根联合体相关联的每个端点在所述第一存储器地址空间中具有相应的地址范围，以及与所述第二主机系统的根联合体相关联的每个端点在所述第二存储器地址空间中具有相应的地址范围。

20. 根据权利要求 19 所述的数据处理系统，其中至少一个端点在所述第一存储器地址空间和所述第二存储器地址空间两者中都具有相应的地址范围。

21. 根据权利要求 13 所述的数据处理系统，其中所述数据处理系统是刀片服务器，以及所述第一主机系统和所述第二主机系统是所述刀片服务器中的刀片。

22. 根据权利要求 13 所述的数据处理系统，其中所述通信架构包括外围组件互连 Express（PCIe）架构，并且所述端点是 PCIe 适配器。

23. 根据权利要求 22 所述的数据处理系统，其中所述 PCIe 架构是包括一个或多个多根感知（MRA）交换机的多根感知 PCIe 架构。

用事务协议和共享存储器在主机系统间通信的系统和方法

技术领域

本申请一般涉及改进的数据处理系统和方法。更具体地，本申请涉及用于使用事务协议和共享存储在主机系统之间通信的机制。

背景技术

大部分现代计算设备利用使用外围组件互连标准的某个版本或实现的总线和输入/输出（I/O）适配器，其中外围组件互连标准最初由英特尔公司在20世纪90年代建立。外围组件互连（PCI）标准规定了用于将外围组件附接到计算机主板的计算机总线。PCT Express或PCIe是使用现有PCI编程概念的PCI计算机总线的实现，但是该实现是基于完全不同且更快的串行物理层通信协议的计算机总线。物理层不包括可以在多个设备之间共享的双向总线，而是包括严格地连接到两个设备的单一单向链路。

图1是示出了根据PCIe规范的PCI Express（PCIe）架构拓扑的示意图。如图1所示，PCIe架构拓扑100包括耦合到根联合体130的主处理器（CPU）100和存储器120，根联合体130接着耦合到一个或多个PCIe端点140（在PCIe规范中使用术语“端点”来表示支持PCIe的I/O适配器）、PCI Express-PCI桥150以及一个或多个互连交换机160。根联合体130表示将CPU/存储器连接到I/O适配器的I/O层级的根。根联合体130包括主桥、集成了零个或多个根联合体的端点、零个或多个根联合体事件收集器、以及一个或多个根端口。每个根端口支持单独的I/O层级。I/O层级可以包括根联合体130、零个或多个互连交换机160和/或桥150（其包含交换或PCIe架构）、以及一个或者多个诸如端点170和182-188的端点。关于PCI和PCIe

的更多信息，参见在外围组件互连特殊兴趣组（PCI-SIG）的网站www.pcisig.com上可获得的PCI和PCIe规范。

今天，将PCI和PCIe I/O适配器、总线等集成到包括刀片服务器的刀片的几乎每一个计算设备的主板上。刀片服务器本质上是用于大量的单独的最小封装的计算机主板“刀片”的壳体，每个刀片包括一个或多个处理器、计算机存储器、计算机存储设备和计算机网络连接，但共享机箱的公共电源和空气冷却资源。刀片服务器对于诸如Web主控和集群计算的特定用途是理想的。

如前所述，通常将PCI和PCIe I/O适配器集成到刀片自身之中。结果，在相同刀片服务器的刀片之间无法共享I/O适配器。而且，I/O适配器的集成限制了链路速率的可扩展性。即，随着时间的过去，链路速率无法随着处理器性能而扩展。到目前为止，还没有设计出一种机制以允许由多个系统镜像跨过多个刀片而共享PCI和PCIe I/O适配器。而且，还没有设计出一种机制以允许以非集成的方式来提供PCI和PCIe I/O适配器，以供刀片服务器中的多个刀片使用。

发明内容

为了解决由于当前PCI和PCIe I/O适配器集成的限制，说明性的实施方式提供了一种机制，其允许由两个或更多的系统镜像（SI）本地地共享PCIe适配器。例如，提供一种机制，用于使在相同根联合体内或跨过多个根联合体（RC）的多个SI能够同时共享例如PCIe I/O适配器的端点，其中所述多个根联合体（RC）共享（即，耦合到）公共PCI交换架构。该机制允许每个根联合体及其相关联的物理和/或虚拟端点（VEP）具有其自身唯一的PCI存储器地址空间。

此外，在基本的PCI规范中缺少但对于管理由端点的共享而产生的联合体配置所需要的是：用于确定和管理在端点中的可能的PCI功能的组合的必要性。因此，此处说明性的实施方式提供了用于在刀片服务器中的第一刀片的一个根联合体与在相同或不同刀片服务器中的第二刀片的第二根联合体进行通信的机制。该说明性的实施

方式通过提供一种机制来支持这种通信，该机制用以初始化在用于支持这种通信的多根刀片集群的根联合体和端点之间的共享存储器。

在一个说明性的实施方式中，多根 PCIe 配置管理器（MR-PCIM）通过发现 PCIe 交换架构（即，PCIe 层级）以及通过遍历所有经由 PCIe 交换架构的互连的交换机可访问的所有链路，来初始化在根联合体和端点之间的共享存储器。因为遍历了链路，MR-PCIM 比较针对根联合体和端点的每个而获得的信息，以确定哪个端点和根联合体位于相同刀片上。然后，生成虚拟 PCIe 树数据结构，该虚拟 PCIe 树数据结构将在 PCIe 交换架构上可用的端点绑定到每个根联合体。作为相同 PCI 树的部分（即，关联于同一根联合体）的端点在虚拟 PCIe 树数据结构中是相关联的。

然后，MR-PCIM 在端点所属的 PCIe 存储器地址空间内对于每个端点给出一个基（base）和界限。类似地，然后，MR-PCIM 在根联合体所属的 PCIe 存储器地址空间内对于每个根联合体给出一个基和界限。为了在各种端点和根联合体的 PCIe 存储器地址空间之间的映射，可以生成存储器转换和保护表数据结构。

例如，对于特定的端点或根联合体，该端点或根联合体可以与第一主机的实际存储器地址空间相关联。通过第二主机，经由第二主存储器上的 PCIe 孔径（aperture），可以访问相同的端点或根联合体，其中该第二主存储器作为直接存储器访问 I/O，通过第一主机的 PCI 总线存储器地址是可访问的。第一主机可以使用存储器转换和保护表数据结构以将由第二主机看到的 PCIe 存储器地址映射到第一主机的实际存储器地址。

在另一个说明性的实施方式中，已经初始化了主机系统的存储器地址空间，使得可以跨过主机系统由根联合体访问端点，然后，可使用这些存储器地址空间以允许与这些根联合体相关联的系统镜像和它们对应的应用来与端点进行通信。

支持这种通信的一种方式是通过队列系统，其中队列系统利用

在不同主机系统中的这些经初始化的存储器地址空间。这种队列系统可包括工作队列结构和完成队列结构。工作队列结构和完成队列结构二者可包括：用于识别大量队列单元（工作队列单元（WQE）或者完成队列单元（CQE），这依赖于该队列结构是工作队列结构还是完成队列结构）的门铃结构、用于队列的起始的基地址、用于队列的末尾的界限地址和指示在队列中将要被处理的下一个 WQE 或 CQE 的偏移量。可以使用工作队列结构和完成队列结构二者来发送和接收数据。

可在对应于将与其进行通信的根联合体和端点的主机系统存储器的部分中提供队列结构和门铃结构。可以生成队列单元，并将所述队列单元添加到队列结构中，以及可以写入门铃结构，以便由此就队列单元可用于处理而通知端点或根联合体。可以执行 PCIe DMA 操作，以取回队列单元以及对应于该队列单元的数据。而且，可以执行 PCIe DMA 操作，以返回完成队列单元（CQE），来指示对队列单元的处理的完成。

根据示例性的一个实施方式，可建立一种面向事务的协议，用于使用说明性的实施方式的共享存储器来在相同或不同的主机系统的根联合体和端点之间进行通信。该面向事务的协议规定了将由例如根联合体或端点的各种单元执行的一系列事务，用以推送或拉回（push or pull）数据。在不脱离本发明的精神和范围的情况下，可利用多种推送和拉回事务的组合。之后，在详细说明中将更详细地描述各种组合。

此外，说明性的实施方式的机制可进一步用于支持在相同或不同主机系统的根联合体和端点之间通过上述的共享存储器而进行的基于套接字协议的通信。利用这种基于套接字的通信，可使用在主机系统中的工作队列以侦听输入的套接字初始化请求。即，希望与第二主机系统建立套接字通信连接的第一主机系统可在其工作队列中生成套接字初始化请求 WQE，并可以向该第二主机系统通知该套接字初始化请求 WQE 对处理可用。

然后，第二主机系统可接受或拒绝该请求。如果第二主机系统接受该请求，则其返回套接字参数的后半个用于由第一主机系统执行在第一和第二主机系统之间的基于套接字的通信中使用。这些参数可以规定队列结构中将要与套接字相关联的部分以及门铃结构，门铃结构用于通知主机系统何时可获得队列单元用于经由套接字进行处理。实际的套接字通信可能涉及例如在主机系统之间拉回事务和/或推送事务。

在根联合体之间的资源的本地共享创建了在 PCIe 架构中的实体和主机系统之间的关系，其可被利用以提供在系统镜像之间和/或端点之间迁移功能及其相关联的应用的机制。需要该迁移功能性，以满足在系统管理领域中对负载均衡能力的不断增长的需要。当前在 PCIe 规范中缺少这种机制。

在一个说明性的实施方式中，单根 PCI 配置管理器（SR-PCIM）提供了具有由端点（EP）支持的可能的虚拟功能（VF）迁移场景的系统镜像。执行管理任务的系统管理员或软件应用（例如，负载均衡应用）可以执行命令，该命令向单根 PCI 管理器（SR-PCIM）指示需要进行从一个 SI 到另一个的对 VF 和与该 VF 相关联的应用的无状态迁移。通过迁移该 VF 及其相关联的应用（其是依赖于 VF 而操作的应用），可补充不同的资源，以在更有效的环境中继续操作。例如，利用负载均衡，可以使用所述说明性的实施方式的机制来移动以太网 VF 及其相关联的依赖的应用，以便利用在不同物理功能（PF）上可用的较快的连接（较少拥塞）的优势，该物理功能（PF）可以与不同的 SI 或甚至 EP 关联在一起。

运行在主机系统上的软件中介（SWI）或虚拟化中介指示 SI 来完成对 VF 的未完成的请求，并且接下来，启动所需的任何进程以将其停止。一旦由 SI 向该 SWI 通知了已经完成了对 VF 的所有请求，则 SWI 可以将与 VF 相关联的任何应用从 SI 移除，并将 VF 从相关联的物理功能（PF）中分离。

然后，SWI 可以将 VF 附加到目标 PF，该 PF 可以在相同或不同

的 EP 中。而且，目标 PF 可以与不同的 SI 相关联。SWI 使 VF 对于现在与该 VF 相关联的 SI 可用，并指令 SI 来配置 VF。SI 配置 VF，由此使其对于相关联的应用可用。然后，SWI 可以指令 SI 来启动相关联的应用，从而它们可以在新迁移的 VF 上使用该资源。

除了上文的机制，说明性的实施方式进一步提供了用于执行将新组件热插入到运行中的多根 PCIe 架构或从运行中的多根 PCIe 架构中热拔出新组件的功能性。这些机制允许根联合体例如热插入到运行中的 PCIe 架构或从运行中的 PCIe 架构中热拔出。例如，可将刀片热插入到刀片机箱，而其相关联的根联合体可以实时地结合到在现有系统中的 PCIe 架构之中。

这种热插/拔能力允许 PCIe 架构增长，并允许跨过新合并的根联合体而本地地共享虚拟功能。因此，可扩展 PCIe 架构而无需为此关闭系统。PCI-SIG I/O 虚拟化标准没有提供用于 CPIe 架构的这种动态扩展的能力或标准。

在一个说明性的实施方式中，提供了一种用于在第一主机系统和第二主机系统之间通信的方法。该方法可以包括初始化第一共享存储器中的第一存储器地址空间，以包括分配给与第二主机系统相关联端点的第一地址范围。该方法还可以包括初始化第二共享存储器中的第二存储器地址空间，以包括分配给与第二主机系统相关联端点的第二地址范围。此外，该方法可以包括在第一地址范围内生成工作队列结构，以及在第二地址范围内生成完成队列结构，该完成队列结构包括第二门铃结构。根据建立的事务协议，第一主机系统和第二主机系统可以执行推送或拉回操作中的至少一个，以在工作队列结构和第二主机系统之间以及在完成队列结构和第一主机系统之间，传送工作队列单元和完成队列单元。

建立的事务协议可以是拉回-拉回-推送事务协议。根据拉回-拉回-推送事务协议，端点执行直接存储器访问操作，以将工作队列单元从工作队列结构拉回到端点。此外，端点执行直接存储器访问操作，以将与工作队列单元对应的数据从第一共享存储器拉回。此外，

端点执行操作以将完成队列单元推送到第一主机系统。

建立的事务协议可以由端点执行。主机系统可通过存储器映射的输入/输出（I/O）操作以及至少一个地址转换和保护表来访问第一地址范围和第二地址范围。推送操作和拉回操作可以是直接存储器访问（DMA）操作。

初始化与第一主机系统相关联的第一存储器地址空间并初始化与第二主机系统相关联的第二存储器地址空间可以包括：遍历数据处理系统的通信架构中的链路，以收集关于通信架构中存在的端点和根联合体的信息。该方法还可以包括生成至少一个虚拟层级，该虚拟层级标识在物理上或逻辑上相互关联的至少一个端点和至少一个根联合体。此外，该方法可以包括基于至少一个虚拟层级来初始化第一存储器地址空间和第二存储器地址空间，使得与第一主机系统的根联合体相关联的每个端点在第一存储器地址空间中具有相应的地址范围，并且与第二主机系统的根联合体相关联的每个端点在第二存储器地址空间中具有相应的地址范围。至少一个端点可以在第一存储器地址空间和第二存储器地址空间两者中都具有相应的地址范围。

数据处理系统可以是刀片服务器，以及第一主机系统和第二主机系统可以是刀片服务器中的刀片。数据处理系统可以包括第一主机系统和第二主机系统与之耦合的外围组件互连 Express（PCIe）架构。端点可以是 PCIe 适配器。PCIe 架构可以是包括一个或多个多根感知（MRA）交换机的多根感知 PCIe 架构。

在另一个说明性的实施方式中，提供了一种计算机程序产品，其包括具有计算机可读程序的计算机可用介质。当在计算设备上执行计算机可读程序时，将使计算设备执行上文关于方法的说明性的实施方式中所概括的各种操作及其组合。

在另一个说明性的实施方式中，提供了一种数据处理系统。该数据处理系统可以包括第一主机系统、第二主机系统以及耦合第一主机系统和第二主机系统的通信架构。数据处理系统还可以执行上

文关于方法的说明性的实施方式中所概括的各种操作及其组合。

在下文对本发明的示例性实施方式的详细描述中，将描述本发明的这些和其他特征及优点，而当看到在下文的对本发明的示例性实施方式的详细描述时，本发明的这些和其他特征及优点对本领域普通技术人员将变得显而易见。

附图说明

在所附的权利要求中阐明了确信新颖的本发明的特征。然而，通过结合附图阅读下文对说明性的实施方式的详细描述，将更好地理解本发明自身、以及优选的使用模式、其进一步的目标和优点，其中：

图 1 是示出了在本领域公知的 PCIe 架构拓扑的示例性示意图；

图 2 是示出了在本领域公知的系统虚拟化的示例性示意图；

图 3 是示出了使用 I/O 虚拟化中介对 PCI 根联合体的 I/O 进行虚拟化的第一方法的示例性示意图；

图 4 是示出了使用本地共享的 PCI I/O 适配器对 PCI 根联合体的 I/O 进行虚拟化的第二方法的示例性示意图；

图 5 是支持 PCIe I/O 虚拟化的端点的示例性示意图；

图 6 是示出了没有本地虚拟化的单根端点的物理和虚拟功能的示例性示意图；

图 7 是示出了支持本地 I/O 虚拟化的单根端点的物理和虚拟功能的示例性示意图；

图 8 是示出了根据一个说明性的实施方式的多根虚拟化 I/O 拓扑的示例性示意图；

图 9 是示出了根据一个说明性的实施方式的从根节点的 SR-PCIM 的视角的多根虚拟化 I/O 拓扑的虚拟层级视图的示例性示意图；

图 10 是示出了根据一个说明性的实施方式的基于共享存储器的 PCIe 系统的示例性示意图；

图 11A 和 11B 是表示了根据一个说明性的实施方式的示例性虚拟 PCI 树数据结构的示意图;

图 12 是概括了根据一个说明性的实施方式的用于为端点的共享而将主机系统的存储器地址空间进行初始化的示例性操作的流程图;

图 13 是示出了根据一个说明性的实施方式的用于从第一主机系统向第二主机系统发送工作队列单元（WQE）的过程的示例性的框图;

图 14 是示出了根据一个说明性的实施方式的用于从第二主机系统向第一主机系统发送完成队列单元（CQE）的过程的示例性的框图;

图 15 是概括了根据一个示例性实施方式的用于在第一主机系统的根联合体和与第二主机系统相关联的端点之间传送 WQE 的示例性操作的示例性流程图;

图 16 是概括了根据一个示例性实施方式的用于从第二主机系统的端点向第一主机系统的根联合体传送 CQE 的示例性操作的示例性流程图;

图 17 是示出了可以用于在相同或不同的主机系统的根联合体和端点之间执行通信的事务的各种可能的组合的示例性表;

图 18 是示出了根据一个示例性实时方式的用于建立套接字并在第一主机系统和第二主机系统之间执行基于套接字的通信的过程的示例性框图;

图 19 是概括了根据一个说明性的实施方式的用于使用基于套接字的通信连接来执行拉回事务的示例性操作的流程图;

图 20 是概括了根据一个说明性的实施方式的用于使用基于套接字的通信连接而执行推送事务的示例性操作的流程图;

图 21A 和 21B 是示出了根据一个说明性的实施方式的从在相同 PCIe 适配器上的一个物理功能向另一个物理功能的虚拟功能及其相关联的应用的单根无状态迁移的示例性示意图;

图 22A 和 22B 是示出了根据一个说明性的实施方式的从一个 PCIe 适配器向另一个 PCIe 适配器的虚拟功能及其相关联的应用的单根无状态迁移的示例性示意图；

图 23A 和 23B 是示出了根据一个说明性的实施方式的从一个系统镜像向另一个系统镜像的虚拟功能及其相关联的应用的单根无状态迁移的示例性示意图；

图 24 是概括了根据一个说明性的实施方式的用于迁移虚拟功能的示例性操作的流程图；

图 25 是示出了根据一个说明性的实施方式的用于根联合体的热插入操作的示例性框图；

图 26 是概括了根据一个说明性的实施方式的用于向 PCIe 架构增加组件的示例性操作的流程图；以及

图 27 是概括了根据一个说明性的实施方式的用于从 PCIe 架构动态移除组件的示例性操作的流程图。

具体实施方式

说明性的实施方式提供了一种机制，该机制允许由相同或不同的根联合体的两个或更多系统镜像（SI）来本地地（natively）共享 PCIe 适配器或“端点”，其中所述相同或不同的根联合体可以位于相同或不同的根节点（例如，刀片服务器的刀片）上。另外，说明性的实施方式提供了一种支持在系统镜像和本地共享的端点之间的通信的机制。此外，说明性的实施方式提供了用于在虚拟平面（plane）、根联合体和系统镜像之间迁移虚拟功能的机制，以实现对 PCIe 架构的管理。另外，说明性的实施方式提供了一种机制，其中通过该机制，根联合体的单根 PCI 管理器（SR-PCIM）能够从端点读取该端点的实现者在设计该端点时所允许的功能的有效组合。然后，SR-PCIM 可以设置将在当前配置中使用的功能的组合，其中在该当前配置中正在使用该端点。

图 2 是示出了本领域中公知的系统虚拟化的示例性示意图。系

统虚拟化是对物理系统的处理器、存储器、I/O 适配器、存储设备以及其他资源的划分，其中每组资源与其自身的系统镜像实例和应用一起独立地操作。在这种系统虚拟化中，虚拟资源由物理资源组成，并作为物理资源的代理来操作，其中物理资源例如为具有相同外部接口和功能的存储器、磁盘驱动器以及具有构建的接口/功能的其他硬件组件。系统虚拟化通常利用虚拟化中介，该虚拟化中介创建虚拟资源并将其映射到物理资源，由此提供虚拟资源之间的隔离。通常，将虚拟化中介提供作为软件、固件和硬件机制之一或其组合。

如图 2 所示，通常在虚拟化系统中，应用 210 与系统镜像（SI）220 进行通信，其中该系统镜像（SI）220 为诸如通用或专用操作系统的软件组件，由该软件组件分配特定的虚拟和物理资源。系统镜像 220 与虚拟系统 230 相关联，虚拟系统 230 包括为运行单个 SI 实例所必需的物理或虚拟化资源，例如，虚拟化的处理器、存储器、I/O 适配器、存储设备等。

系统镜像 220 通过使用虚拟系统 230 而经由虚拟化中介 240 来访问物理系统资源 250。虚拟化中介 240 管理对 SI 的资源分配，并隔离分配给 SI 的资源免受其他 SI 访问。通常，基于由虚拟化中介 240 执行的资源映射以及由虚拟化中介 240 维护的一个或多个资源映射数据结构来执行这种分配和隔离。

可使用这种虚拟化以允许对 I/O 操作和 I/O 资源的虚拟化。即，关于 I/O 虚拟化（IOV），可由使用 I/O 虚拟化中介（IOVI）（诸如虚拟化中介 240）的多于一个的 SI 来共享单个物理 I/O 单元。IOVI 可以是软件、固件等，用于通过干预例如一个或多个的配置、I/O、来自 SI 的存储器操作、以及直接存储器访问（DMA）、完成和对 SI 的中断操作来支持 IOV。

图 3 是示出了使用 I/O 虚拟化中介的对 PCI 根联合体的 I/O 进行虚拟化的第一方法的示例性示意图。如图 3 所示，主机计算机组 310 可以是一个或者多个芯片处理器、主板、刀片等，该主机计算机组 310 可以支持多个系统镜像 320-330，应用（未示出）通过这些系统

镜像可以访问诸如 PCIe 端点 370-390 的系统资源。通过 I/O 虚拟化中介 340、PCIe 根联合体 350 以及一个或多个 PCIe 交换机 360 和/或其他 PCIe 架构单元，该系统镜像与虚拟化的资源进行通信。

通过图 3 所示的方法，I/O 虚拟化中介 340 介入到所有的 I/O 事务中，并执行所有的 I/O 虚拟化功能。例如，I/O 虚拟化中介 340 将来自各种 SI 的 I/O 队列多路传输到 PCIe 端点 370-390 中的单一队列。这样，I/O 虚拟化中介充当在 SI 320-330 和物理 PCIe 端点 370-390 之间的代理。

这种 I/O 虚拟化中介 340 的介入可能在 I/O 操作中引入额外的延迟，这限制了每时间单位的 I/O 操作的数量，并且由此限制了 I/O 性能。此外，I/O 中介需要额外的 CPU 周期，这样，降低了对其他系统操作可用的 CPU 性能。此方法所需要的额外的上下文交换和中断重定向机制也会影响系统的整体性能。而且，当在多个根联合体之间共享端点 370-390 时，IOVI 340 是不可行的。

图 4 是示出了使用本地共享的 PCI I/O 适配器对 PCI 根联合体的 I/O 进行虚拟化的第二方法的示例性示意图。如图 4 中所示，主机处理器组 410 可以是一个或多个芯片处理器、主板、刀片等，该主机处理器组 410 可以支持多个系统镜像 420-430，应用（未示出）通过这些系统镜像可以访问诸如 PCIe I/O 虚拟化（IOV）端点 470-490 的系统资源。该系统镜像 420-430 通过 PCIe 根联合体 440 和一个或多个 PCIe 交换机 460、和/或其他 PCIe 架构单元来与虚拟化的资源进行通信。

PCIe 根联合体 440 包括根联合体虚拟化引擎（enabler）（RCVE）442，其中该根联合体虚拟化引擎（RCVE）442 可包括一个或多个地址转换和保护表数据结构、中断表数据结构等，其实现与支持 IOV 的端点 470-490 的 I/O 操作的虚拟化。例如，可以由 PCIe 根联合体 440 使用地址转换和保护表数据结构来执行在用于虚拟化资源的虚拟和实际地址之间的地址转换，基于虚拟资源至 SI 的映射来控制对虚拟资源的访问，以及其他虚拟化操作。例如，通过 PCIe 存储器地

址空间可访问这些根联合体中断表数据结构，并且这些根联合体中断表数据结构可用于将中断映射到与 SI 相关联的合适的中断处理器。

如图 3 所示的布置，在图 4 的虚拟化结构中也提供了 I/O 虚拟化接口 450。将 I/O 虚拟化接口 450 与不支持 IOV 的 PCIe 端点一起使用，其中所述不支持 IOV 的 PCIe 端点可耦合到 PCIe 交换机 460。即，对于那些对 I/O 虚拟化（IOV）没有本地（即，在端点内部）支持的 PCIe 端点，以同如前所述的关于图 3 相类似的方式，将 I/O 虚拟化接口（IOVI）450 与 PCIe 端点一起使用。

对于支持 IOV 的 PCIe 端点 470-490，使用 IOVI 450 主要用于配置事务的目的，并且在存储器地址空间操作中不涉及 IOVI 450，所述存储器地址空间操作诸如为从 SI 发起的存储器映射的输入/输出（MMIO）操作、或者从 PCIe 端点 470-490 发起的直接存储器访问（DMA）操作。相反，直接执行在 SI 420-430 和端点 470-490 之间的数据传输，而无需由 IOVI 450 干预。如同将在下文更详细描述的，通过 RCVE 442 和支持 IOV 的 PCIe 端点 470-490 的内置 I/O 虚拟化逻辑（例如，物理和虚拟功能），在 SI 420-430 和端点 470-490 之间的直接 I/O 操作变为可能。执行直接 I/O 操作的能力极大地增加了能够执行 I/O 操作的速度，但这需要 PCIe 端点 470-490 支持 I/O 虚拟化。

图 5 是支持 PCIe I/O 虚拟化（IOV）的端点的示例性示意图。如图 5 中所示，PCIe IOV 端点 500 包括 PCIe 端口 510，通过该端口可执行与 PCIe 架构的 PCIe 交换机等的通信。内部路由 520 提供到配置管理功能 530 和多个虚拟功能（VF）540-560 的通信通路。配置管理功能 530 可以是与虚拟功能 540-560 相对的物理功能。如同在 PCI 规范中所用的，术语物理“功能”是由单一配置空间所表示的一组逻辑。换言之，物理“功能”是电路逻辑，其基于在存储器中与该功能相关联的配置空间中存储的数据是可配置的，例如可在不可分离的资源 570 中提供。

可使用配置管理功能 530 来配置虚拟功能 540-560。在支持 I/O 虚拟化的端点内，虚拟功能是共享一个或多个例如链路的物理端点资源，并且可以与其他功能一起提供在例如 PCIe IOV 端点 500 的可共享资源池 580 中的功能。无需通过 I/O 虚拟化中介的运行时干预，虚拟功能可以直接是针对来自系统镜像的 I/O 和存储器操作的宿（sink），以及是对系统镜像（SI）的中断、完成、以及直接存储器访问（DMA）操作的源。

PCIe 端点关于由 PCIe 端点所支持的“功能”可以具有许多不同的配置类型。例如，端点可以支持单物理功能（PF）、多个独立的 PF、或甚至多个依赖的 PF。在支持本地 I/O 虚拟化的端点中，由端点支持的每个 PF 可以与一个或多个虚拟功能（VF）相关联，这些虚拟功能（VF）自身可以依赖于与其他 PF 相关联的 VF。将在下文的图 6 和图 7 中示出在物理和虚拟功能之间示例性关系。

图 6 是示出了没有本地虚拟化的单根端点的物理和虚拟功能的示例性示意图。术语“单根端点”是指与单根节点（即，单主机系统）的单根联合体相关联的端点。利用单根端点，可由与单根联合体相关联的多个系统镜像（SI）共享该端点，但无法在相同或不同的根节点上的多个根联合体之间共享该端点。

如图 6 所示，根节点 600 包括：与 PCIe 端点 670-690 通信的多个系统镜像 610、612；I/O 虚拟化中介 630（其如前所述地使用）；PCIe 根联合体 640;以及一个或多个 PCIe 交换机 650 和/或其他 PCIe 架构单元。根节点 600 进一步包括单根 PCIe 配置管理（SR-PCIM）单元 620。SR-PCIM 单元 620 负责管理 PCIe 架构和端点 670-690，该 PCIe 架构包括根联合体 640、一个或多个 PCIe 交换机 650 等。SR-PCIM 620 的管理责任包括确定要将哪个功能分配给哪个 SI 610、620，并建立端点 670-690 的配置空间。根据 SI 的能力以及来自用户（诸如，系统管理员）的输入、或者关于将哪些资源分配给哪个 SI 610、612 的负载均衡软件，SR-PCIM 620 可以配置各种端点 670-690 的功能。SI 的能力可以包括各种因素，这些因素包括：多少地址空

间可用于分配给端点 670-690，多少中断可用于分配给端点 670-690 等等。

每个 PCIe 端点 670-690 可以支持一个或多个物理功能（PF）。一个或多个 PF 可以彼此独立，或以某种方式彼此依赖。基于供应商定义的功能依赖性，PF 可以依赖于另一个 PF，其中例如一个 PF 需要另一个 PF 的操作或者由另一个 PF 生成的结果，以便正确地操作。在所描述的例子中，PCIe 端点 670 支持单 PF，而 PCIe 端点 680 支持 1 到 M 的不同类型的多个独立的 PF（即，PF0 到 PFN）。类型涉及 PF 或 VF 的功能性，例如以太网功能和光纤通道功能是两种不同类型的功能。端点 690 支持具有两个或多个相关 PF 的不同类型的多个 PF。在所描述的例子中，PF0 依赖于 PF1，或者反之亦然。

在图 6 中示例性的例子中，端点 670-690 是由系统镜像（SI）610-612 通过由 I/O 虚拟化中介（IOVI）630 可用的虚拟化机制而共享的。如前所述，在这种布置中，在 SI 610、612 和 PCIe 端点 670-690 之间的所有 PCIe 事务中涉及 IOVI 630。单独的 PCIe 端点 670-690 无需在其自身中支持虚拟化，这是因为处理虚拟化的负担完全放在 IOVI 630 上。结果，虽然在这种布置中可以使用已知的用于虚拟化的机制，但与如果在每个 I/O 操作中均没有涉及 IOVI 630 的 I/O 速率潜力相比，可以执行 I/O 操作的速率相对较慢。

图 7 是示出了支持本地虚拟化的单根端点的物理和虚拟功能的示例性示意图。在图 7 中示例性的布置与图 6 的布置相似，但由于 PCIe 端点 770-790 本地地（即，在端点自身内部）支持 I/O 虚拟化（IOV）而有一些重要不同。结果，针对支持 IOV 的 PCIe 端点 770-790，可以有效地移除在图 6 中的 I/O 虚拟化中介 630，当然，不能移除配置操作。然而，如果在此布置中还利用了不支持 IOV 的 PCIe 端点（未示出），例如，传统端点，则可以连同在图 7 中示例性的单元来使用 I/O 虚拟化中介，以处理在系统镜像 710 和 712 之间的对这种不支持 IOV 的 PCIe 端点的共享。

如图 7 中所示，支持 IOV 的 PEIe 端点 770-790 可以支持一个或

多个独立或依赖的物理功能（PF），然后该物理功能（PF）可以与一个或多个独立或依赖的虚拟功能（VF）相关联。在此上下文中，由 SR-PCIM 720 使用 PF 来管理一组 VF，以及也使用 PF 管理诸如物理错误和事件的端点功能。与 PF 相关联的配置空间定义了 VF 的能力，包括与该 PF 相关联的 VF 的最大数量、PF 和 VF 与其他 PF 和 VF 的组合等。

由 SI 使用 VF 来访问位于支持 IOV 的 PCIe 端点 770-790 上的资源，例如存储器空间、队列、中断等。这样，针对将要共享特定 PF 的每个 SI 710 和 712 来生成不同的 VF。在对应的 PF 的配置空间中，由端点 770-790 基于 SR-PCIM 720 的 VF 的数量的设置来生成 VF。按照这种方式，将 PF 虚拟化，使得可由多个 SI 710、712 来共享该 PF。

如图 7 中所示，VF 和 PF 可以依赖于其他 VF 和 PF。通常，如果 PF 是依赖的 PF，那么与该 PF 相关联的所有 VF 也将是依赖的。这样，例如，PF0 的 VF 可以依赖于对应的 PF1 的 VF。

对于图 7 所示的布置，SI 710、712 可以通过 PCI 根联合体 730 和 PCIe 交换机 740 与支持 IOV 的 PCIe 端点 770-790 直接进行通信，反之亦然，而无需包括 I/O 虚拟化中介。通过在端点 770-790 中和在 SR-PCIM 720 中提供的 IOV 支持，可以进行这种直接通信，SR-PCIM 720 对在端点 770-790 中的 PF 和 VF 进行配置。

在 SI 和端点之间的直接通信显著地增加了可以在多个 SI 710-712 和共享的支持 IOV 的 PCIe 端点 770-790 之间执行 I/O 操作的速度。然而，为使这种性能增强变得可行，PCIe 端点 770-790 必须通过在 SR-PCIM 720 和端点 770-790 的物理功能（PF）中提供用于生成和管理虚拟功能（VF）的机制而支持 I/O 虚拟化。

上文对 PCIe 层级的描述局限于单根层级。换言之，仅由在与单 PCI 根联合体 730 相关联的单根节点 700 上的 SI 710、712 来共享 PCIe 端点。上述的机制对共享 PCIe 端点的多根联合体没有提供支持。这样，无法向多个节点提供对 PCIe 端点的资源的共享访问。这限制了

利用这种布置的系统的可扩展性，这是因为对于每个根节点需要单独的端点集合。

这里，说明性的实施方式利用多个根 I/O 虚拟化，其中多个 PCI 根联合体可以共享对同一组支持 IOV 的 PCIe 端点的访问。结果，与这些 PCI 根联合体的每个相关联的系统镜像的每个可以共享对同一组支持 IOV 的 PCIe 端点资源的访问，但在适当位置处具有针对在每个根节点上的每个 SI 的虚拟化的保护。这样，通过提供允许添加根节点和对应的 PCI 根联合体的机制而将可扩展性最大化，其中这些根联合体可以共享支持 IOV 的 PCIe 端点的相同的现有组。

图 8 是示出了根据一个说明性的实施方式的多根虚拟化的 I/O 拓扑的示例性示意图。如图 8 所示，提供多个根节点 810 和 820，每一个根节点具有单根 PCI 配置管理器（SR-PCIM）812、822、一个或多个系统镜像（SI）814、816、824、826、以及 PCI 根联合体 818 和 828。将这些例如可以是在刀片服务器中的刀片的根节点 810 和 820 耦合到 PCIe 交换架构的一个或多个多根感知（MRA）PCIe 交换机 840，其中该 PCIe 交换架构可以包括一个或多个这种 MRA PCIe 交换机 840 和/或其他 PCIe 架构单元。MRA 交换机 840 不同于在图 7 中的非 MRA 交换机的类型，原因在于 MRA 交换机 840 具有用于附加的根节点的连接，并包含用于保持那些不同根节点的地址空间独立和独特所需要的机制。

除了这些根节点 810 和 820，还提供包括多根 PCI 配置管理器（MR-PCIM）832 和对应的 PCI 根联合体 834 的第三根节点 830。MR-PCIM 832 是负责发现并配置图 8 中示出的在多根（MR）拓扑中的虚拟层级，这将在下文更加详细地描述。这样，MR-PCIM 832 针对多根节点的多根联合体匹配端点的物理和虚拟功能。SR-PCIM 812 和 822 配置与其相关联的单根联合体的物理和虚拟功能。换言之，MR-PCIM 将 MR 拓扑看作一个整体，而 SR-PCIM 仅看到在 MR 拓扑内的其自身的虚拟层级，这将在下文更加详细地描述。

如图 8 所示，支持 IOV 的 PCIe 端点 850 和 860 支持一个或多个

虚拟端点（VE）852、854、862、864。VE 是分配给根联合体的一组物理和虚拟功能。这样，例如在支持 IOV 的 PCIe 端点 850 和 860 上为根节点 810 的 PCI 根联合体 818 提供单独的 VE 852 和 862。类似地，在支持 IOV 的 PCIe 端点 850 和 860 上为根节点 820 的 PCI 根联合体 828 提供单独的 VE 854 和 864。

将每个 VE 分配给具有单根联合体的虚拟层级（VH），在该层级中，单根联合体作为 VH 的根，而 VE 作为终结节点。VH 是分配给根联合体或 SR-PCIM 的完整功能的 PCIe 层级。应该注意，将 VE 中的所有物理功能（PF）和虚拟功能（VF）分配给相同的 VH。

每个支持 IOV 的 PCIe 端点 850 和 860 支持基本功能（BF）859 和 869。BF 859、869 是由 MR-PCIM 832 所使用的物理功能，用于管理相应的端点 850、860 的 VE。例如，BF 859、869 负责向相应端点 850、860 的 VE 分配功能。MR-PCIM 832 通过使用在 BF 的配置空间中的字段而向 VE 分配功能，该配置空间允许将 VH 号分配给在端点 850、860 中的每个 PF。尽管本发明并非局限于此，在所说明性的实施方式中，每个节点仅有一个 BF。

如图 8 中所示，每个 VE 852、854、862 和 864 可支持其自身的物理和虚拟功能组。如前所述，这种功能组可以包括独立的物理功能、依赖的物理功能、以及它们的相关的独立/依赖虚拟功能。如图 8 中所示，VE 852 利用其相关联的虚拟功能（VF）来支持单物理功能（PF0）。VE 854 同样地利用其相关联的虚拟功能（VF）来支持单物理功能（PF0）。VE 862 支持多个独立的物理功能（PF0-PFN）以及其相关联的虚拟功能（VF）。然而，VE 864 支持多个依赖的物理功能（PF0-PFN）。

当且仅当将 VE 分配给 SI 已经访问的 VH 时，VE 852、854、862 或 864 可以与根节点 810 和 820 的 SI 814、816 和 826 直接通信，并且反之亦然。端点 850 和 860 自身必须支持诸如前面所述的单根 I/O 虚拟化，以及如关于当前说明性的实施方式而描述的多根 I/O 虚拟化。这种要求所基于的事实在于：拓扑支持多根联合体，但每个单

独的根节点仅看到其相关联的基于单根的虚拟层级。

图 9 是示出了根据一个说明性的实施方式的从根节点的根联合体的视角所见的多根虚拟化 I/O 拓扑的虚拟层级视图的示例性示意图。如图 9 中所示，虽然多根（MR）拓扑可以是如图 8 所示的那样，但是每个单独的根节点的每个根联合体仅看到它的 MR 拓扑的部分。这样，例如，与根节点 810 相关联的 PCI 根联合体 818 看到它的主机处理器组、它自己的系统镜像（SI）814、816、MRA 交换机 840、以及它自己的虚拟端点（VE）852 和 862。在此虚拟层级中，存在完全的 PCIe 功能性，然而，PCI 根联合体 818 没有看到不是它自身的虚拟层级部分的 VE、根联合体、系统镜像等。

由于此布置，在 MP 拓扑中的根节点的根联合体之间的通信上施加了限制。即，因为将 PCIe 的功能性局限于与根联合体相关联的虚拟层级，所以根联合体无法与另一个根联合体通信。而且，与各种根联合体相关联的系统镜像无法与其他根联合体的系统镜像通信。为解决这种限制，这里说明性的实施方式提供了各种机制，用以对在虚拟层级之间（具体地是在不同根节点的根联合体之间）的通信提供支持。

对于说明性的实施方式的主机系统，为了经由其根联合体与多个端点通信，该主机系统使用由各种端点和根联合体共享的共享存储器，其中该主机系统与该根联合体相关联。为了确保端点与主机系统正确的操作，必须初始化该共享存储器，使得与主机系统相关联的每个端点被提供有其自己的共享存储器部分，其中通过该共享存储器可以执行各种通信。说明性的实施方式利用了用于初始化主机系统的共享存储器的机制，其中发现 PCIe 架构，并且将 PCIe 架构的端点虚拟地绑定到该主机系统的根联合体。然后，为每个端点和根联合体给出每个主机系统的共享存储器地址空间的每个端点和根联合体自己的部分，每个端点和根联合体是虚拟地绑定到每个主机系统。通过主机系统的共享存储器的这些部分，与一个主机系统的根联合体相关联的端点可以同其他主机系统的一个或多个其他根

联合体进行通信。

图 10 是示出了根据一种说明性的实施方式的基于共享存储器 PCIe 的系统的示例性示意图。如图 10 所示，系统 1000 具有包括第一根联合体（RC1）1012 和第一端点（EP1）1014 的主机系统 1010，并且主机系统 1010 与第一多根感知（MRA）交换机 1016 相关联，该多根感知（MRA）交换机 1016 可以同样作为主机系统 1010 的一部分而提供。系统 1000 具有包括第二根联合体（CR2）1022 和第二端点（EP2）1024 的第二主机系统 1020，并且第二主机系统 1020 也与第二多根感知（MRA）交换机 1026 相关联，该多根感知（MRA）交换机 1026 可以同样作为主机系统 1020 的一部分而提供。这些主机系统 1010 和 1020 的每一个可以代表例如在相同的多根刀片集群系统 1000 中的单独的刀片。可选地，可以在单独的计算设备上完全地提供主机系统 1010 和 1020。每个主机系统 1010 和 1020 位于其自己的虚拟层级（VH）中。通过与 PCIe 架构 1030 的一个或多个 MRA 交换机 1016、1026 和 1032 的通信链路，主机系统 1010 和 1020 彼此连接，并且与其他共享端点 EP3-EP6 1042-1044 和 1052-1054 相连接。与主机系统 1010 和 1020 以及端点 1042-1044 和 1052-1054 相关联的通信链路可以与一个或多个虚拟平面（VP）相关联。

在 PCIe 架构中没有使用虚拟层级（VH）标识符以区分哪个主机系统 1010 和 1020 与给定的 PCIe 事务相关联。作为替代，使用了链路本地虚拟平面（VP）标识符。由于 VP 标识符是链路本地的，所以 RC 1 的 VH 可以是例如在 1032 和 1016 之间的链路上具有 VP=4，而在 1032 和 1042 之间的链路上具有 VP=4。换言之，VH 由一组 PCIe 组件和连接这些组件的链路构成，这些链路的每个都具有链路本地 VP 标识符，用于指明给定事务正引用哪个 VH。

在所描述的例子中，目标是允许根联合体 1012、以及因此允许与同该根联合体 1012 相关联的一个或多个系统镜像相联合运行的应用来同与另一个根联合体相关联的端点（例如，与根联合体 RC2 1022 相关联的端点 EP2 1024）进行通信。这样，例如，可由运行在根联

合体 RC1 1012 上的系统镜像来将 EP2 1024 作为端点而使用。按照这种方式，可以在不同虚拟平面和/或主机系统上的系统镜像之间共享与根联合体位于同一位置的端点。结果，当在节点之间通信时，可以实现高性能的节点到节点（即，主机系统到主机系统）的通信和负载均衡，并通过消除对通过诸如 InfiniBand 或以太网交换机的外部网络适配器和交换机的需要而降低系统成本。

为了允许在主机系统之间由系统镜像共享端点，在主机系统 1010 或 1020 之一或者单独的主机系统 1060 中提供的多根 PCI 配置管理器（MR-PCIM）1062 初始化主机系统的存储器空间 1070 和 1080，以建立用于根联合体和端点的基（base）和界限孔径（limit aperture）。MR-PCIM 1062 通过 PCIe 架构 1030 中的一个或多个 MRA 交换机 1032 和 MRA 交换机 1064 来访问 PCIe 架构 1030。

MR-PCIM 1062 通过各种互连的交换机，以本领域公知的方式遍历（traverse）PCIe 架构 1030 的链路，以识别与 PCIe 架构 1030 相关联的根联合体和端点。然而，对于说明性的实施方式所执行的遍历，除了执行发现架构遍历操作的根联合体（RC）之外，在该发现架构遍历期间，将所有的根联合体（RC）视为端点。

当 MR-PCIM 1062 遍历 PCIe 架构时，它在根联合体和端点之间执行大量检查，以确定给定的根联合体与给定的端点是否相关联。根据产生的信息，MR-PCIM 1062 生成一个或多个虚拟 PCI 树数据结构，其将在 PCIe 架构 1030 上可用的端点绑定到每个根联合体。在虚拟 PCI 树数据结构中，与相同根联合体相关联的端点之间是互相关联的。

当 MR-PCIM 1062 发现并配置了架构后，相应的 RC 允许它们相关联的 SR-PCIM 1018 和 1028 发现并配置 VH。每个 SR-PCIM 1018、1028 为每个给定的端点分配在其所属的 PCIe 存储器地址空间中的基地址和界限，该 PCIe 存储器地址空间例如是与主机系统 1 存储器 1070 和主机系统 2 存储器 1080 相关联的 PCIe 存储器地址空间。SR-PCIM 1018、1028 将所述基地址和界限写入 EP 的基地址寄存器

（BAR）。然后，可将工作请求和完成消息写入 PCI 存储器地址空间的这些部分，以便实现在跨过主机系统 1010 和 1020 的不同根联合体和端点之间的通信，这将在下文更加详细地描述。

如上所述，对于说明性的实施方式，当 MR-PCIM 1062 遍历 PCIe 架构 1030 时，它在根联合体和端点之间执行大量的检查。例如，如 PCI 规范所定义的，MR-PCIM 1062 访问每个功能（EP 的物理功能和虚拟功能）的 PCIe 配置空间，其中 PCIe 配置空间位于 EP 中。例如，MR-PCIM 也访问对于每个端点的重要产品数据（VPD）字段，并为稍后的比较而存储 VPD 信息，诸如存储在耦合到 MR-PCIM 1062 的非易失性存储器区域（未示出）中。

VPD 是唯一地定义了诸如系统的硬件、软件和微码单元等项目的信息。VPD 向系统提供了关于各种字段可替换单元（FRU）的信息，其中字段可替换单元（FRU）包括供应商名称、零件编号、序列号和对经营、资产管理和任何需要 PCI 设备唯一标识的事情有用的其他详细信息。VPD 信息通常位于 PCI 设备（诸如端点 1014、1024）的存储设备（例如串行 EEPROM）内。可以从在 www.pcisig.com 可获得的 PCI 本地总线规范 3.0 版本来获得关于 VPD 的更多信息。

在已经取回并存储了对于每个端点 1014、1024、1042、1044、1052 和 1054 的 VPD 信息之后，MR-PCIM 1062 识别哪些 EP 和 RC 驻留在例如刀片的同一硬件设备上。例如，MR-PCIM 1062 访问包含共同驻留（co-residency）字段的 MRA 交换机 1016、1026、1032 的 VPD 信息，所述共同驻留字段指示它与保持 RC 和 EP 的硬件设备相关联。MRA 交换机 1016、1026、1032 存储分配给 RC 的 VH，然后，可以使用该 VH 以确定哪些 EP 和 RC 驻留在相同的硬件设备上。

在确定了 EP 与 RC 共同存在相同的主机上之后，MR-PCIM 1062 创建一个或多个虚拟 PCI 树数据结构，诸如图 11A 和 11B 中所示例性的。如同在图 11A 和 11B 中示例性的，虚拟 PCI 树数据结构将在 PCIe 架构上可用的端点绑定到每个根联合体。

假设在图 11A 中示例性的虚拟 PCI 树数据结构中，通过由用户

指示给 MR-PCIM 1062 的分配，使得端点 EP2 1024、EP4 1044 和 EP5 1052 与根联合体 RC1 1012 相关联。仅执行上述的 VPD 匹配以允许 RC 确定 EP 物理地位于该 RC 的主机上。这告诉 RC，通过在 RC 的地址空间中的标准存储器映射寻址，EP 对该 RC 是可访问的。这是物理的关联。利用虚拟 PCI 树数据结构，通过用户指示他/她希望 MR-PCIM 1062 创建这种逻辑关联来指定逻辑关联。

类似地，在图 11B 中假设端点 EP1 1014、EP3 1042 和 EP6 1054 通过描述逻辑关联的用户输入、和它们的 VPD 信息以及由 MR-PCIM 1062 做出的比较，来与根联合体 RC1 1012 相关联。这样，在图 11A 中示例性的所描述的例子中，端点 EP2 1024 经由交换机 2 1026 和交换机 1 1016 关联于（或绑定到）根联合体 RC1 1012。端点 EP4 1044 和 EP5 1052 经由交换机 3 1032 和交换机 1 1016 而与根联合体 RC1 1012 相关联。在图 11B 中，端点 EP1 1014 经由交换机 1 1016 和交换机 2 1026 而关联于（或绑定到）根联合体 2 1022。端点 EP3 1042 和 EP6 1054 经由交换机 3 1032 而与根联合体 RC2 1022 相关联。

基于这些虚拟 PCI 树数据结构，MR-PCIM 1062 对每个端点在其所属的 PCIe 存储器地址空间内分配基地址和限制。可将基地址存储于端点的基地址寄存器（BAR）中。例如，通过两个 PCIe 存储器地址空间 1070 和 1080 可访问 EP1 1014。在主机系统 1 1010 中，通过主机系统的存储器 1070 地址空间，该主机系统的处理器（未示出）可访问 EP1 1014。在主机系统 2 1020 中，EP1 1014 具有在主机系统 2 的存储器 1080 地址空间中的由 EP1 基地址和限制定义的 PCIe 孔径，其中主机系统 2 的存储器 1080 地址空间可通过 PCI 总线存储器地址，经由存储器影射的 I/O 来访问。例如，主机系统 1 1010 的处理器可使用存储器地址转换和保护表（未示出），以将由主机系统 2 1020 的处理器看到的 PCIe 存储器地址映射成为主机系统 1 的存储器地址，该存储器地址转换和保护表诸如是在虚拟化中介（诸如管理程序、根联合体 1012 等）中提供的。

类似地，通过用于主机系统存储器 1070 和 1080 的两个 PCI 存

储器地址空间，可访问 EP2 1024。在主机系统 2 1020 中，由主机系统 2 的处理器通过用于其存储器 1080 的主机系统 2 的实际存储器地址可访问 EP2 1024。在主机系统 1 1010 中，EP2 1024 具有在主机系统 1 的存储器 1070 中的由用于 EP2 1024 的基地址和限制所定义的 PCIe 孔径，其中作为存储器映射的 I/O，通过 PCI 总线存储器地址可访问该存储器 1070。主机系统 2 1020 可使用存储器地址转换和保护表（未示出），以将由主机系统 1 1010 看到的 PCIe 存储器地址映射到主机系统 2 的实际存储器地址。

可针对根联合体 RC1 1012 和 RC2 1022 来初始化主机系统存储器 1070 和 1080 的类似部分。例如，在主机系统 1 1010 中，由主机系统 1 的处理器，通过用于主机系统 1 的存储器 1070 的主机系统 1 的实际存储器地址可访问 RC1 1012。RC1 1012 在主机系统 2 的存储器空间中具有 PCIe 孔径，其中经由直接存储器访问（DMA）I/O，通过主机系统 1 的 PCI 总线存储器地址可访问该主机系统 2 的存储器空间。主机系统 1 1010 可使用存储器地址转换和保护表(未示出)，以将由主机系统 2 1020 看到的 PCIe 存储器地址映射成为主机系统 1 的实际存储器地址。

类似地，在主机系统 2 1020 中，主机系统 2 的处理器通过用于存储器 1080 的主机系统 2 的实际存储器地址可访问 RC2 1022。RC2 1022 具有在主机系统 1 的存储器 1070 中的 PCIe 孔径，其中作为 DMA I/O，通过主机系统 2 的 PCI 总线存储器地址可访问该主机系统 1 的存储器 1070。主机系统 2 1020 可使用存储器地址转换和保护表（未示出），以将由主机系统 1 1010 看到的 PCIe 存储器地址映射成为主机系统 2 的实际存储器地址。

这样，说明性的实施方式的机制提供了对在主机系统中的存储器空间的初始化，使得在多个主机系统中，可由多于一个的根联合体来访问端点。然后，可由根联合体利用分配给不同端点的存储器空间的部分，以向端点发送请求和完成消息，和从端点发送请求和完成消息。

图 12 是概括了根据一个说明性的实施方式的用于为端点的共享而初始化主机系统的存储器地址空间的示例性操作的流程图。应该理解，可以由计算机程序指令来实现在图 12 中示例性的流程图和下文描述的流程图的每一块以及在流程图中的块的组合。可以向处理器或其他可编程数据处理装置提供这些计算机程序指令，以产生一种机器，使得在处理器或其他可编程数据处理装置上执行的指令创建用于实现在一个或多个流程图块中描述的功能的装置。也可以在计算机可读存储器或存储介质中存储这些计算机程序指令，该计算机可读存储器或存储介质能引导处理器或其他可编程数据处理装置以特定方式运行，使得在计算机可读存储器或存储介质中存储的指令产生包括指令装置的制造物品，其中所述指令装置实现了在一个或多个流程图块中描述的功能。

这样，流程图的块支持用于执行所描述的功能的装置的组合、用于执行所描述的功能的步骤的组合、以及用于执行所描述的功能的程序指令装置。还应该理解，流程图的每个块以及流程图中的块的组合可由基于专用硬件的计算机系统或专用硬件和计算机指令的组合实现，该基于专用硬件的计算机系统执行所描述的功能或步骤。

如图 12 所示，操作开始于 MR-PCIM 通过遍历所有链路而发现该 PCIe 架构（步骤 1210），其中所述链路通过 PCIe 架构的互连的交换机可访问。存储在 PCIe 架构的发现期间所发现的对于每个端点和根联合体的 VPD 信息（步骤 1220）。

MR-PCIM 将对于每个端点的 VPD 信息与对于每个根联合体的 VPD 信息进行比较，以确定给定的端点与给定的根联合体是否相关联（步骤 1230）。对于每次比较，如果对于端点和根联合体的 VPD 信息相匹配，则 MR-PCIM 设置对应的共同驻留字段（步骤 1240）。基于所发现的端点和根联合体信息以及针对每个比较的共同驻留字段的设置，MR-PCIM 生成一个或多个虚拟 PCI 树数据结构（步骤 1250）。

基于所生成的虚拟 PCI 树数据结构，MR-PCIM 为每个端点在该

端点所属的每个 PCIe 存储器地址空间内分配基地址和界限（步骤1260）。基于所生成的虚拟 PCI 树数据结构，MR-PCIM 为每个根联合体在根联合体所属的每个 PCIe 存储器地址空间内分配基地址和界限（步骤 1270）。然后，该操作终止。

已经初始化了主机系统的存储器地址空间，使得根联合体跨过主机系统可访问端点，然后，可使用这些存储器地址空间，以允许与这些根联合体相关联的系统镜像和它们对应的应用来与端点进行通信。实现这种通信的一种方式是通过队列系统，该队列系统利用在不同主机系统中的这些经初始化的存储器地址空间。这种队列系统可包括工作队列结构和完成队列结构。工作队列结构和完成队列结构二者可包括用于识别大量队列单元（或者是工作队列单元（WQE）或者是完成队列单元（CQE），这依赖于该队列结构是工作队列结构还是完成队列结构）的门铃结构、用于队列的起始的基地址、用于队列的末尾的界限地址、和指示在队列中将要处理的下一个 WQE 或 CQE 的偏移量。可以使用工作队列结构和完成队列结构二者来发送和接收数据。

图 13 是示出了根据一个说明性的实施方式的用于从第一主机系统向第二主机系统发送工作队列单元（WQE）的过程的示例性的框图。出于描述的目的，假设建立了具有主机系统的系统，所述主机系统例如是通过 PCIe 架构 1330 连接的第一主机系统 1310 和第二主机系统 1320，所述主机系统具有多个共享的 PCI 根联合体，例如，RC1 1312 和 RC2 1322，其中第一主机系统 1310 和第二主机系统 1320 可以包括 MRA 交换机 1316 以及 1326。进一步假设：位于具有根联合体 RC2 1322 的第二主机系统 1320 中的端点（例如 EP2 1324）将跨过 PCIe 架构 1330 而与第一主机系统 1310 的根联合体 RC1 1312 共享，并且，将该端点恰当地映射到第二主机系统 1320 的内部存储器 1380 的地址空间以及第一主机系统的 PCI 总线存储器地址空间。可通过例如使用前面关于例如图 10 至图 12 所描述的初始化机制来实现该系统配置。

如图 13 所示，通过与两个主机系统存储器 1370 和 1380 相关联的存储器空间，可访问端点 EP1 1314。在第一主机系统 1310 上，第一主机系统的处理器通过用于第一主机系统的存储器 1370 的第一主机系统的实际存储器地址可访问端点 EP1 1314。在第二主机系统 1320 上，端点 EP1 1314 具有在第二主机系统的存储器 1380 上的 PCIe 孔径 1382，其中作为存储器映射的 I/O，通过 PCI 总线存储器地址可访问该第二主存储器 1380。第一主机系统 1310 可使用存储器地址转换和保护表（ATPT）1318，以将由第二主机系统 1320 看到的 PCIe 存储器地址映射成为用于第一主机系统的存储器空间 1370 的实际存储器地址。

类似地，通过两个主机系统存储器空间 1370 和 1380 可访问端点 EP2 1324。在第二主机系统 1320 中，第二主机系统的处理器通过第二主机系统的实际存储器地址和存储器地址空间 1380 可访问端点 EP2 1324。在第一主机系统 1310 中，端点 EP2 1324 具有在第一主机系统的存储器 1370 上的 PCIe 孔径 1372，其中作为存储器映射的 I/O，通过 PCI 总线存储器地址可访问该第一主机系统的存储器 1370。第二主机系统 1320 可使用存储器地址转换和保护表（ATPT）1328，以将由第一主机系统 1310 发送的 PCIe 存储器地址映射到第二主机系统的存储器空间 1380 的实际存储器地址。

工作队列结构 1374 可包括门铃结构 1375，该门铃结构 1375 用以传递大量的 WQE、用于队列的开始的基地址、用于队列的末尾的界限地址以及指示在工作队列中将要处理的下一个 WQE 的偏移量。类似地，完成队列结构 1376 可包括门铃结构 1377，该门铃结构 1377 用于传递大量的 CQE、用于队列的开始的基地址、用于队列的末尾的界限地址、以及指示在完成队列中将要处理的下一个 CQE 的偏移量。

为了从第一主机系统 1310 向第二主机系统 1320 发送 WQE，第一主机系统 1310 通过向其发送工作队列 1374 插入一个或多个 WQE 而发起该过程。每个 WQE 包含数据段的列表，其中每个数据段包括

都位于第二主机系统的PCIe存储器总线地址空间中的基地址和界限地址，并且还通过地址转换和保护表（ATPT）将该基地址和界限地址映射到在第一主机系统的存储器空间1370中的实际存储器地址。

然后，第一主机系统1310将所发送的WQE的数量写入到用于门铃结构1375的端点EP2的PCIe地址之中。通过ATPT，将用于此门铃结构的地址映射到第一主机系统的PCIe存储器总线地址空间，并且还映射到在第二主机系统的存储器空间1380中的实际存储器地址。当门铃写操作完成时，第二主机系统1320的RC轮询或者收到一个中断然后轮询，以通过第一主机系统的实际存储器地址空间1380取回门铃结构1375。即，可将第二主机系统1320的RC配置为周期地轮询用于门铃结构1375的地址，以确定是否要处理新的WQE。可选地，第一主机系统1310对门铃结构1375的设置可生成对第二主机系统1320的中断，以向第二主机系统1320的RC通知可获得新的WQE用于处理。然后，第二主机系统1320的RC可以轮询用于新的WQE信息的门铃结构1375，并且据此处理它们。

然后，端点EP2 1324对根联合体RC1 1312执行PCIe DMA操作以取回WQE。每个DMA操作使用第一主机系统的PCIe存储器总线地址空间，并将DMA操作的结果放入第二主机系统的存储器1380，其中在第二主机系统1320上，通过其实际存储器地址空间可访问第二主机系统的存储器1380。这样，使用主机系统1310和1320的初始化的共享存储器，实现了在不同主机系统1310和1320中的根联合体和端点之间的工作队列单元的通信。

图14是示出了根据一个说明性的实施方式的用于从第二主机系统1320向第一主机系统1310发送完成队列单元（CQE）的过程的示例性的框图。如图14所示，一旦完成了与一个WQE或一组WQE相关联的工作，则端点EP2 1324对根联合体RC1 1312执行一个或多个PCIe DMA操作，以向根联合体RC1 1312发送一个或多个CQE。在RC1 1312可以轮询或等待指示CQE可用的中断的意义上，可以使用门铃。

每个DMA操作使用第一主机系统的PCIe存储器总线地址空间，并将结果放入第一主机系统1310上的存储器1370，其中在第一主机系统1310上，通过其实际存储器地址空间可访问第一主机系统的存储器1370。优选地，将结果存储在存储器1370的DMA可寻址部分，依赖于所使用的特定OS，DMA可寻址部分位于存储器1370中的不同部分。

图15是概括了根据一个示例性实施方式的用于在第一主机系统的根联合体和与第二主机系统相关联的端点之间传送 WQE 的示例性操作的示例性流程图。如图15所示，该操作开始于第一主机系统向其发送工作队列插入一个或多个WQE（步骤1510）。然后，第一主机系统将所发送的 WQE 的数量写入对于门铃结构的目标端点的PCIe地址之中（步骤1520）。当门铃写操作完成时，第二主机系统轮询或者收到中断然后轮询，以通过第一主机系统的实际存储器地址空间来取回门铃结构（步骤1530）。

然后，目标端点对第一主机系统的根联合体执行 PCIe DMA 操作，以取回WQE（步骤1540）。然后，目标端点将DMA操作的结果放入第二主机系统的存储器（步骤1550）。然后，该操作终止。

图16是概括了根据一个示例性实施方式的用于从第二主机系统的端点向第一主机系统的根联合体传送CQE的示例性操作的示例性流程图。该操作开始于端点完成与向端点提交的一个或多个WQE相关联的处理工作（步骤1610）。然后，该端点对与主机系统相关联的根联合体执行一个或多个PCIe DMA操作，以向根联合体发送一个或多个 CQE，其中从该主机系统接收该一个或多个 WQE（步骤1620）。将DMA操作的结果放入第一主机系统的存储器（步骤1630）。然后，该操作终止。

这样，可使用说明性的实施方式的共享存储器，以提供队列结构，其中通过该队列结构，可在不同主机系统上的根联合体和端点之间交换工作请求和完成消息。这样，根联合体可以同与提供了该根联合体的主机系统不同的主机系统上的端点进行通信，并且反之

亦然。

根据在此示例性的一种实施方式，可建立一种面向事务的协议，用于使用说明性的实施方式的共享存储器，以在相同或不同的主机系统的根联合体和端点之间通信。如同下文将要描述的，面向事务的协议描述了将由例如根联合体或端点的各种单元执行的一系列事务，以推送或拉回数据。

返回图 13，上文关于向端点提供 WQE 并向根联合体返回 CQE 的方式的描述是拉回-拉回-推送协议的一个例子。即，响应于第一主机系统 1310 对门铃结构 1375 的写入，第二主机系统 1320 的端点 EP2 1324 使用 PCIe DMA 操作从第一主机系统的共享存储器 1370 拉回 WQE。这些 WQE 提供了用于将要执行的操作的“命令”。基于在 WQE 中存储的段信息，第二主机系统 1320 的端点 EP2 1324 从在第一主机系统的共享存储器 1370 的工作队列结构 1374 中拉回对应的数据。一旦完成了对应于 WQE 的工作，则第二主机系统 1320 的端点 EP2 1324 使用一个或多个 PCIe DMA 操作向第一主机系统 1310 的根联合体 RC1 1312 推送 CQE。这样，在上文描述的图 13 的例子中利用了拉回-拉回-推送事务协议。

拉回和推送事务的其他可能的组合对于不同的事务协议的建立也是可能的。图 17 是示出了可以用于执行在相同或不同主机系统的根联合体和端点之间的通信的事务的多种可能组合的示例性表。如图 17 所示，可以连同说明性的实施方式的机制一起来利用拉回和推送事务的任何组合，以便由此建立事务协议，用于对说明性的实施方式的共享存储器的使用。

根联合体和端点负责实施选择的协议。例如，OS 系统栈和端点执行操作，用于拉回和推送数据，作为诸如如前所述的选择的事务协议的部分。对将要利用的协议的选择依赖于由端点所利用的特定的 PCIe 架构，例如 InfiniBand 或以太网架构。可以根据编程的选择，例如是否使用轮询、中断处理或轮询和中断处理的组合，来确定协议的特殊性。

说明性的实施方式的机制可进一步用于支持基于套接字协议的通信，该通信是通过上述的共享存储器来在相同或不同主机系统的根联合体和端点之间进行。当存在恒定的连接时，可使用这种套接字协议。可基于所希望的效率和可靠性来确定是使用套接字协议还是基于事务的协议，诸如，上文描述的推送-拉回事务。

利用套接字协议，可使用在主机系统中的工作队列，以侦听输入的套接字初始化请求。即，希望与第二主机系统建立套接字通信连接的第一主机系统可在其工作队列中生成套接字初始化请求WQE，并向该第二主机系统通知该套接字初始化请求 WQE 对处理可用。然后，第二主机系统可接受或拒绝该请求。如果第二主机系统接受该请求，它返回套接字参数的后一半，用于由第一主机系统在第一和第二主机系统之间执行基于套接字的通信中使用。这种通信可以涉及例如在主机系统之间的拉回事务和/或推送事务。

图 18 是示出了根据一个说明性的实施方式的用于在第一主机系统和第二主机系统之间建立套接字并执行基于套接字的通信的过程的示例性的框图。在说明性的实施方式的基于套接字的实现中，在主机系统（诸如主机系统 1810）上的端点（诸如 EP2 1824）包含接收缓冲器 1876、缓冲器满标记 1877 以及门铃结构 1878。缓冲器满标记 1877 和门铃结构 1878 可包括用以指示事件已经发生的存储器中的地址。例如主机系统 1810 的发送者主机系统通过在接收者主机系统 1820 的存储器 1870 中的 PCIe 孔径 1872 来写入门铃结构 1878，其中 PCIe 孔径 1872 可由发送者主机系统的根联合体 RC1 1812 访问，对应于例如端点 EP2 1824 的连接端点。

如前所述，在为实现在相同或不同的主机系统上的多个根联合体之间共享端点而初始化主机系统的共享存储器期间，针对每个所发现的根联合体和端点来读出重要产品数据（VPD）信息，以便生成虚拟 PCI 树数据结构。该 VPD 信息可包括指示特定的根联合体或端点是否支持 PCIe 上的套接字的字段。根据一种说明性的实施方式，可使用此信息以标识可以与哪些端点建立套接字用于基于套接字的

通信。

这样，在初始化期间，第一主机系统 1810 可以例如通过在用于端点 EP2 1824 的 VPD 中的供应商特定字段、可由如前所述的 MR-PCIM 以及由主机系统自身可访问的位于 EP 中的 VPD 信息，来确定端点 EP2 1824 支持 PCIe 上的套接字。类似地，第二主机系统 1820 可通过在对于端点 EP1 1814 的 VPD 信息中的其供应商特定字段来确定端点 EP1 1814 支持 PCIe 上的套接字。

每个主机系统 1810 和 1820 具有工作队列（WQ）1850 和 1860，该工作队列用于监听输入的套接字初始化请求。例如，第二主机系统 1820（即，接收主机系统）阻塞或等待到其工作队列 1860 表面的套接字初始化请求，或拉回端点 EP2 1824 的门铃结构 1878，以确定套接字初始化请求是否已经到达。套接字初始化请求包含到工作队列 1850 中的基、界限和起始偏移量，这将用于套接字的第一主机系统的那一半。

第一主机系统 1810（即，发送主机系统）可在其工作队列 1850 中生成套接字初始化请求，并可以写入 EP2 1824 的门铃结构 1878，指示套接字初始化请求 WQE 可用。一旦在门铃结构 1878 中取回数据时，第二主机系统的端点 EP2 1824 可执行 PCIe DMA 操作，以使用根联合体 RC1 的 PCIe 总线存储器地址来从第一主机系统的工作队列 1850 中取回套接字初始化请求，端点 EP2 1824 可访问该根联合体 RC1 的 PCIe 总线存储器地址。

然后，第二主机系统 1820 可解析该套接字初始化请求，并且以应用或操作系统特定的方式确定接受还是拒绝该套接字初始化请求。如果第二主机系统 1820 拒绝套接字初始化请求，则第二主机系统 1820 向第一主机系统的根联合体 RC1 1812 发送非连接响应 PCIe DMA，并且如果需要，则中断第一主机系统的根联合体 RC1 1812。

如果第二主机系统 1820 接受套接字初始化请求，则端点 EP2 1824 对第一主机系统的根联合体 RC1 1812 执行 PCIe DMA 操作，指示套接字参数的后一半，即，基、界限和在工作队列 1860 内的起始

偏移量，用于套接字的第二主机系统的那一半。

一旦已经以上述方式初始化了套接字，可按照拉回事务或推送事务两种方式之一，使用建立的套接字来执行发送/接收操作。利用拉回事务，第一主机系统 1810 的根联合体 RC1 1812 通过向其工作队列 1850 写入 WQE 而执行发送操作，并然后写入与端点 EP2 1824 相关联的门铃结构 1878，其中通过根联合体 RC1 1812 PCIe 总线存储器地址空间可访问该门铃结构 1878。当门铃写入操作完成时，第二主机系统或者 1820 轮询或者收到中断然后轮询，以通过第二主机系统的实际存储器地址空间来取回门铃结构 1878。端点 EP2 1824 然后对根联合体 RC1 1812 执行 PCIe DMA 操作，以取回与发送操作相关联的 WQE。PCIe DMA 操作使用第一主机系统的 PCIe 存储器总线地址空间，并将结果放入第二主机系统上的存储器 1880，其中通过第二主机系统的实际存储器地址空间可访问该存储器 1880。第二主机系统 1820 然后取回在 WQE 中描述的并且与发送操作相关联的数据段。

当第二主机系统完成在 WQE 中的工作请求时，端点 EP2 1824 对根联合体 RC1 1812 执行 PCIe DMA 操作，以推送信令通知发送操作已经完成的 CQE。该 DMA 操作使用第一主机系统的 PCIe 存储器总线地址空间，并将结果放入第一主机系统上的存储器 1870，其中通过第一主机系统的实际存储器地址空间可访问该存储器 1870。

对于推送事务，根联合体 RC2 1822 写入用于端点 EP1 1814 的门铃结构 1888，指示已经可用的接收 WQE 的数量。当端点 EP1 1814 有数据要发送时，端点 EP1 1814 检查以确定端点 EP1 1814 在根联合体 RC2 1822 的工作队列 1860 上是否具有可用的接收 WQE。如果没有可用的接收 WQE，则根联合体 RC1 1812 写入端点 EP2 的缓冲器满标记 1887，以指示第一主机系统 1810 有数据要在套接字上发送，而第二主机系统 1820 针对该套接字需要通过接收 WQE 来公告一些缓冲区。

如果存在可用的接收 WQE，则第二端点 EP2 1824 对根联合体

RC1 1812 执行 PCIe DMA 操作，以取回在根联合体 RC1 的工作队列 1850 中下一个可用的 WQE。DMA 操作使用第一主机系统的 PCIe 存储器总线地址空间，并将结果放入在第二主机系统 1820 上的存储器 1880 中，其中通过第二主机系统的实际存储器地址空间可访问该存储器 1880。然后，第二主机系统 1820 将其数据发送到在接收 WQE 中传递的数据段。

当第二主机系统 1820 完成工作请求时，端点 EP2 1824 然后对根联合体 RC1 1812 执行 PCIe DMA 操作，以推送信令通知发送操作已经完成的 CQE。该 DMA 操作使用第一主机系统的 PCIe 存储器总线地址空间，并将结果放入第一主机系统 1810 上的存储器，其中通过第一主机系统的实际存储器地址空间可访问该存储器。

图 19 是概括了根据一个说明性的实施方式的用于使用基于套接字的通信连接来执行拉回事务的示例性操作的流程图。该操作开始于第一主机系统的根联合体向其工作队列写入 WQE（步骤 1910），并然后写入与目标端点相关联的门铃结构（步骤 1920）。当门铃写入操作完成时，第二主机系统轮询或者收到中断然后轮询，以通过第二主机系统的实际存储器地址空间来取回门铃（步骤 1930）。

目标端点然后对第一主机系统的根联合体执行 PCIe DMA 操作，以取回与发送操作相关联的 WQE（步骤 1940）。目标端点将 PCIe DMA 操作的结果放入在第二主机系统上的存储器（步骤 1950）。第二主机系统然后取回在 WQE 中描述的并且与发送操作相关联的数据段（步骤 1960）。

响应于第二主机系统完成在 WQE 中请求的工作（步骤 1970），目标端点对第一主机系统的根联合体执行 PCIe DMA 操作，以推送信令通知发送操作已经完成的 CQE（步骤 1980）。第一主机系统的根联合体将 PCIe DMA 操作的结果放入第一主机系统的存储器（步骤 1990）。然后，该操作终止。

图 20 是概括了根据一个说明性的实施方式的用于使用基于套接字的通信连接来执行推送事务的示例性操作的流程图。第二主机系

统的根联合体写入用于第一主机系统的端点的门铃结构，指示根联合体已经可用的接收 WQE 的数量（步骤 2010）。响应于第一主机系统的端点具有要发送的数据（步骤 2020），第一主机系统的端点检查以确定该端点在第二主机系统的根联合体的工作队列上是否具有任何可用的接收 WQE（步骤 2030）。如果没有可用的接收 WQE，则第一主机系统的根联合体将第二主机系统的缓冲器满标记写入第二端点，以指示第一主机系统有数据要在套接字上发送，并且第二主机系统需要针对该套接字将接收 WQE 记入一些缓冲器（步骤 2040）。然后，操作返回到步骤 2030。

如果存在可用的接收 WQE，第二端点对第一主机系统的根联合体执行 PCIe DMA 操作，以取回在第一主机系统的工作队列的根联合体上可用的下一个 WQE（步骤 2050）。第二端点将 PCIe DMA 操作的结果放入在第二主机系统上的存储器（步骤 2060）。然后，第二主机系统将其数据发送到在接收 WQE 中传递的数据段（步骤 2070）。

当第二主机系统完成了所请求的工作时，第二端点对第一主机系统的根联合体执行 PCIe DMA 操作，以推送信令通知发送操作已经完成的 CQE（步骤 2080）。第二端点将 PCIe DMA 操作的结果放入在第一主机系统上的存储器（步骤 2090）。然后，该操作终止。

如同上文所讨论的，多根系统的端点可以支持具有一个或多个相关联的虚拟功能的一个或多个物理功能。说明性的实施方式的机制，除了在相同或不同的主机系统的根联合体和端点之间提供通信，也提供用于管理端点的物理和虚拟功能的机制。由说明性的实施方式的机制提供的一种功能提供了将单根无状态虚拟功能及其相关联的应用从一个物理功能迁移到相同端点上的另一个的能力。对于满足在系统管理领域对负载均衡能力的不断增长的需要来说，该迁移功能性是重要的。

通过迁移 VF 及其相关联的应用（该应用是依赖于 VF 而操作的应用），可补充不同的资源，以在更有效的环境中继续操作。例如，

利用负载均衡，可以使用说明性的实施方式的机制来移动以太网 VF 及其相关联的依赖的应用，以便利用在不同 PF 上可用的较快（例如，较少拥塞）连接的优势，其中 PF 可以与不同的 SI 或甚至 EP 相关联在一起。

图 21A 和 21B 是示出了根据一个说明性的实施方式的从在相同端点（例如，PCIe 适配器）上的一个物理功能向另一个物理功能的虚拟功能及其相关联的应用的单根无状态迁移的示例性示意图。如图 21A 中所示，如连接单元 2110 和 2120 的虚线所示，与系统镜像（SI）2105 相关联的应用 2110 与虚拟功能（VF）2120 相关联。基于来自 SR-PCIM 2100 的信息，软件中介 2115 可以向系统管理员或者等同的管理负责人来描述迁移场景。这可以包括但不限于显示在 PCIe 架构中可用的等同的 VF，该 VF 可以是用于经由系统管理接口（未示出）迁移的目标。

可以例如基于 VF 迁移能力位来确定可由 SR-PCIM 2100 描述的特定的迁移场景，其中 SR-PCIM 访问该 VF 迁移能力位以确定特定的 VF 是否可迁移。基于来自 SR-PCIM 2100 的所述信息，SWI 2115 可以通过管理控制台或实体解译并将此数据转换为对用户可用的 VF 迁移场景。这些迁移场景将高度依赖于所讨论的组件的设计。例如，为了迁移以太网适配器，OS 可能必须对其进行解配置。如果 OS 没有提供这种功能性，则管理实用工具将无法描述这种场景。换言之，管理实用工具维护组件（系统镜像类型、硬件等）的知识，然后使用该知识来描述迁移场景。所述信息以及存储于 VF 的迁移能力位中的迁移能力信息标识哪些用于迁移的场景可供选择。

系统管理员启动该过程，以迁移所希望的 VF 2120 和相关联的应用 2110。例如，诸如在图形用户界面显示中，管理软件（未示出）可将 VF 及其相关联的应用描述为实体，其中可在主机系统和 PCIe 架构上的可用资源之间迁移所述实体。管理软件可以存在于诸如从国际商业机器公司可获得的 HMC 的硬件管理控制台上、或者存在于设计为与固件（例如，软件中介或管理程序）交互并控制硬件资源

的功能的系统运行软件的任何其他控制台或部分之中。

运行在主机系统上的软件中介（SWI）2115 可向 SI-A 2105 发送请求，以请求完成所有的未完成的请求或迁移 VF2120 的灵活性，其中软件中介（SWI）2115 是任意类型的固件或软件代码，其使用在管理应用和硬件之间以创建允许额外的功能性的抽象层。例如，SI-A 2105 和 SWI 2115 可以具有应用程序接口（API），SI-A 2105 和 SWI 2115 通过应用程序接口（API）进行通信。SI-A 2105 可以通过暂停或停止使用 VF 2120 的任何应用 2110 来响应该请求。SI-A 2105 可以确保完成对 VF 2120 的未完成的所有请求。本质上，SI-A 2105 检查以确信所有的队列都处于表示没有未决的请求以及已经完成了所有事务的状态中。例如，完全此任务的一种方式是检查所有的 WQE 都具有对应的 CQE。

然后，SI-A 2105 可以对其 VF 2120 的逻辑表示进行解配置，有效地停止 SI-A 对 VF 2120 的使用。这是可以由例如用于在 SI-A 2105 上的 VF 2120 的设备驱动器（未示出）执行的操作。然后 SI-A 2105 可以向 SWI 2115 通知所有请求已经完成以及可以移除 VF 2120。SWI 2115 接着可以从 SI-A 2105 中移除 VF 2120。这将使 VF 2120 呈现出由 SI-A 2105 不可探测和不可配置。SWI 2115 现在可以通过清除在端点的配置空间中的 VF 的表示来将 VF 2120 从目标物理功能（PF）2135 中分离。

现在参考图 21B，然后，SWI 2115 可以将目标 VF 2145 附接到其 PF 2140。然后 SWI 2115 使得 VF 2145 可用于由 SI-A 2105 进行配置，并指示 SI-A 2105 配置 VF 2145。例如，SWI 2115 更新在固件中的 SI-A 的设备树，以包括例如 VF 2145 的新设备，该 VF 2145 可以呈现为例如新的端点。该新端点或 VF 2145 可以是任何类型的端点设备，其在 OS 中的逻辑表示依赖于在 SI-A 的设备树中发现它的设备驱动器，其中由固件代码将该设备树提供给 OS。一旦在 SI-A 的设备树中存在用于新设备（例如 VF 2145）的实体，则用于该设备的设备驱动器将检测并配置该新设备。

一旦 SI-A 2105 使用例如设备驱动器配置 VF 2145，则相关联的应用 2110 能够使用 VF 2145。现在，SWI 2115 现在可指示 SI-A 2105 启动完成该迁移的相关联的应用 2110。结果，如由虚线所示，应用 2110 和 VF 2120 依然是相关联的，但已经将 VF 2120 从其与 PF 2135 相关联迁移到现在的与 PF 2140 相关联。

图 22A 和图 22B 是示出了根据一个说明性的实施方式的从例如 PCIe 适配器的一个端点向另一个端点的对虚拟功能（VF）及其相关联的应用的单根无状态迁移的示例性示意图。用于从一个端点向另一个的 VF 的无状态迁移的操作类似于上文关于图 21A 和图 21B 的描述。总而言之，在图 21A 至图 21B 的操作和图 22A 至图 22B 的操作之间的主要的不同在于：VF 位于不同的端点上，而不是仅与在相同端点内的不同的物理功能相关联。

如图 22A 中所示，如由连接单元 2210 和 2220 的虚线所示的，与系统镜像（SI）2205 相关联的应用 2210 是与虚拟功能（VF）2220 相关联。SR-PCIM 2200 向系统管理员或等同的管理负责人描述迁移场景。这可以包括但不限于显示在 PCIe 架构上可用的等同的 VF，其中所述 VF 可以是用于经由系统管理接口（未示出）的迁移的目标。

系统管理员启动该过程，以迁移所希望的 VF 2220 和相关连的应用 2210。例如，管理软件（未示出）可将 VF 及其相关联的应用描述为诸如在管理控制台或实体的图形用户界面显示中的实体，其中可在 PCIe 架构和主机系统上的可用资源之间迁移所述实体。运行在主机系统上的软件中介（SWI）2215 可以向 SI-A 2205 发送针对待迁移的 VF 2220 完成所有未完成的请求的请求。例如，SI-A 2205 和 SWI 2215 可以具有应用程序接口（API），其中 SI-A 2205 和 SWI 2215 通过该应用程序接口（API）进行通信。SI-A 2205 可以通过暂停或停止使用 VF 2220 的任何应用 2210 来响应该请求。SI-A 2205 可以确保完成对 VF 2220 的所有未完成的请求。

然后，SI-A 2205 可以解配置其 VF 2220 的逻辑表示，有效地停止 SI-A 对 VF 2220 的使用。这是可以由例如用于在 SI-A 2205 上的

VF 2220 的设备驱动器（未示出）执行的操作。然后，SI-A 2205 可以向 SWI 2215 通知所有请求已经完成以及可以移除 VF 2220。SWI 2215 接着可以从 SI-A 2205 中移除 VF 2220。这将使 VF 2220 呈现出由 SI-A 2205 不可探测和不可配置。SWI 2215 现在可以通过清除在端点的配置空间中的 VF 的表示来将 VF 2220 从目标物理功能（PF）2235 中分离。

现在参考图 21B，然后，SWI 2215 可以将目标 VF 2245 附接到其 PF 2240 上，该 PF 2240 位于与 PF 2235 不同的端点上，其中 VF 2220（现在是 VF 2245）最初是与 PF 2235 相关联的。然后，SWI 2215 使 VF 2245 可用于 SI-A2205 进行配置，并且指示 SI-A 2205 配置 VF 2245。例如，SWI 2215 更新在固件中的 SI-A 的设备树，以包括新设备。SI-A 2205 可以使用例如设备驱动器来配置 VF 2245，设备驱动器的类型将依赖于所讨论的设备或功能的特定属性。相关联的应用 2210 现在可以使用 VF 2245。SWI 现在可以指示 SI-A 2205 启动完成迁移的相关联的应用 2210。结果，如由虚线所示，应用 2210 和 VF 2220 依然是相关联的，但已经将 VF 2220 从其与 PF 2235 相关联迁移到现在的与在不同端点上的 PF 2240 相关联。

可以执行类似的操作，以从一个系统镜像向另一个系统镜像迁移虚拟功能。图 23A 和图 23B 是示出了根据一个说明性的实施方式的从一个系统镜像向另一个系统镜像的对虚拟功能及其相关联的应用的单根无状态迁移的示例性图。如图 23A 所示，用于停止目标为待迁移的 VF 2320 的操作的操作与前面关于图 21A 和图 22A 的描述基本上相同。一旦停止了与 VF 2320 相关联的应用 2310，并且完成了目标为 VF 2320 的操作，则 SI-A 2305 解配置其 VF 2320 的逻辑表示，向 SWI 2314 通知已经完成所有请求并且可以移除 VF 2320。

如果针对 VF 2320 将执行 SI 改变，则 SWI 2315 将 VF 2320 从相关联的 PF 2335 分离，并将 VF 2345 附接到目标 PF 2340。目标 PF 2340 可以位于相同或不同的端点上。SWI 2315 使得 VF 2345 可用于例如 SI-B 2350 的目标 SI 进行配置，并且指示目标 SI 2350 配置 VF

2345。目标 SI 2350 有效地配置 VF 2345，使其可为相关联（现在与 SI-B 2350 相关联）的应用 2310 所用。SWI 2315 通知目标 SI 2350 启动相关联的应用，以使用在新的 VF 2345 上的资源。

图 24 是概括了根据一个说明性的实施方式的用于迁移虚拟功能的示例性操作的流程图。如图 24 所示，操作开始于用户指定将要迁移的 VF、以及针对该 VF 的目标目的地（步骤 2410）。运行在主机系统上的 SWI 向 SI 发送完成对 VF 的所有未完成的请求以便迁移 VF 的请求（步骤 2420）。SI 暂停或停止使用 VF 的任何应用（步骤 2430），并确保已经完成了对 VF 的所有未完成的请求（步骤 2440）。然后，SI 解配置其 VF 的逻辑表示（步骤 2450）。SI 向 SWI 通知已经完成所有请求并且可以移除 VF（步骤 2460）。

然后，SWI 从 SI 中移除 VF，并将 VF 从相关联的 PF 分离（步骤 2470）。然后，SWI 将 VF 附接到可以在相同或不同端点上并且可以与相同或不同的系统镜像相关联的的目标 PF（步骤 2480）。然后，SWI 指示现在与 VF 相关联 SI 来配置 VF，由此使其可为相关联的应用所用（步骤 2490）。SWI 指示 SI 启动相关联的应用，以使用在新 VF 上的资源（步骤 2495）。然后，该操作终止。

这样，利用说明性的实施方式，可在相同的端点内、不同的端点之间、以及在相同或不同端点上的不同的系统镜像之间迁移虚拟功能。这种迁移使得可以执行多种负载均衡操作。而且，这种迁移允许将虚拟功能移动到更有益于虚拟功能的有效操作的操作环境。

这样，概括的说明性的实施方式提供了用于在相同的根联合体内或跨过多个根联合体（RC）的多个系统镜像（SI）之间同时共享例如 PCIe 适配器的端点的机制。而且，说明性的实施方式的机制支持使用基于队列的通信、基于推送-拉回协议的通信和基于套接字的通信的能力。另外，说明性的实施方式提供了用于从在相同或不同端点上一个物理功能向另一个物理功能以及从一个系统镜像向另一个系统镜像迁移虚拟功能及其相关联的应用实例的机制。

除了这些机制，说明性的实施方式进一步提供了用于执行将新

组件热插入到运行中的多根 PCIe 架构或从运行中的多根 PCIe 架构中热拔出的功能性。这些机制允许根联合体例如热插入到运行中的 PCIe 架构或从运行中的 PCIe 架构中热拔出。例如，可将刀片热插入到刀片机箱，而其相关联的根联合体可以实时地结合到在现有系统中的 PCIe 架构之中。

这种热插/拔能力允许 PCIe 架构扩展并且在新合并的根联合体之间本地地共享虚拟功能。因此，PCIe 架构可扩展，而无需为此停下系统。PCI-SIG I/O 虚拟化标准没有提供这种用于 CPIe 架构的动态扩展的能力或标准。

利用所说明性的实施方式机制，假设存在具有一个或多个 PCI 根联合体和支持多根感知（MRA）的交换机的现有主机系统。例如，主机系统可以具有两个根联合体 RC1 和 RC2，其中根联合体 RC1 和 RC2 由具有一个或多个 MRA 交换机的 PCI 架构连接。而且，假设存在一个或多个端点连接到该 PCIe 架构的端点，其中可将该 PCIe 架构配置为与现有的根联合体以及新引入的根联合体进行通信。而且，假设多根（MR）PCI 配置管理器（MR-PCIM）能够并且已经通过遍历所有通过互连的 PCIe 架构的交换机可访问的链路而发现 PCIe 架构，其中该多根（MR）PCI 配置管理器（MR-PCIM）可位于主机系统之一的带内或带外。上文描述的关于在此阐明的说明性的实施方式的多种机制可满足所有这些假设。

利用上文假设的配置，当系统管理员等向现有的 PCIe 架构添加新的根联合体（例如，向刀片机箱中插入新的刀片）时，自动机制（诸如热插入控制器）或系统管理员至少之一通过诸如管理员接口等向 MR-PCIM 通知根联合体的添加。例如，可通过向 MR-PCIM 公告一个指示已经发生了向架构添加新实体的动作的事件来进行这种通知。这种事件可标识交换机和交换机端口，其中在所述交换机和交换机端口处现在新的根联合体连接到 PCIe 架构，即，在所述交换机和交换机端口处插入了根联合体。

然后，MR-PCIM 可以通过执行大量操作以初始化在现有 PCIe

架构中的新根联合体而处理该公告的事件。例如，MR-PCIM 可利用关于新增加的组件的信息来更新其 PCIe 架构配置数据结构。由 MR-PCIM 使用 PCIe 架构配置数据结构来表示 PCIe 架构的配置。由 MR-PCIM 从 PCIe 架构配置寄存器并从来自系统管理员的输入（例如，通过与 MR-PCIM 的管理用户接口）来收集存储在 PCIe 架构配置数据结构中的信息。下面，将更充分地描述 PCIe 架构配置数据结构的内容以及对此内容的使用。

在更新了 PCIe 架构配置数据结构之后，然后，MR-PCIM 执行如由 PCI 规范所定义的 PCI 配置空间操作，以按照 PCI 规范来确定新增加的组件的特征，例如，是否是端点、根联合体、交换机等，端点、根联合体、交换机等是何种类型等等。如果确定新增加的组件是交换机，则关于交换机的每个端口来执行 PCI 配置空间操作，以确定存在耦合到交换机的附加组件。然后，将针对新增加的组件的特征信息（例如组件类型、供应商名称、零件编号、序列号等）存储在虚拟 PCIe 架构配置数据结构中，以供 MR-PCIM 使用。

如果组件是新根联合体或新端点，MR-PCIM 将该新根联合体或端点与虚拟平面相关联。按照这种方式，使得该新根联合体或端点对系统可用。如果组件是 MRA 交换机，则 MR-PCIM 按照 PCI I/O 虚拟化规范来配置该交换机的虚拟平面表。如果组件是交换机，则检查交换机端口，以察看哪些组件（如果有的话）附接到这些端口，并且 MR-PCIM 也按照类似的方式基于关于这些组件的信息来配置其 PCIe 架构配置数据结构。按照这种方式，可以动态地将新组件添加到 PCIe 架构。

关于新根联合体，已经将 MR-PCIM 配置为包括对于新根联合体的特征信息，并将该新根联合体与虚拟平面相关联，可以在 PCIe 架构中使用该新联合体。这样，说明性的实施方式的机制允许向现有运行中的 PCIe 架构添加根联合体及其相关联的组件。结果，当扩展系统以包括附加组件时，针对该系统没有停机时间。

图 25 是示出了根据一个说明性的实施方式的用于根联合体的热

插入操作的示例性框图。应该理解，尽管图 25 示出了用于根联合体的热插入操作，但示例性实施方式并不局限于此。相反，正如上文所述的，该热插入操作可以关于端点、交换机以及其他类型的组件来执行而并不脱离本发明的精神和范围。

如图 25 中所示，主机系统 2510 具有分别与虚拟平面 2511 和 2513 相关联的现有根联合体 RC1 2512 和 RC2 2514。现有根联合体 RC1 2512 和 RC2 2514 由 MR-PCIM 2562 配置以与在虚拟平面 2541 和 2550 内的端点 2542、2544、2552 以及 2554 进行通信。MR-PCIM 2562 维护架构配置数据结构 2566，其中架构配置数据结构 2566 存储了用于 PCIe 架构 2530 的所有组件以及附接到 PCIe 架构 2530 的组件（包括主机系统 2510 的组件）的特征信息。

在所描绘的例子中，假设已经将新根联合体 RC N 2516 加入主机系统 2510。例如，该根联合体 RC N 2516 可以与刀片相关联，并且主机系统 2510 可以是具有机箱的刀片服务器，其中可以插入与 RC N 2516 相关联的刀片。可以使用可提供 RC N 2516 的其他类型的设备，而不脱离说明性的实施方式的精神和范围。

对于说明性的实施方式，存在与每个交换机端口相关联的两种类型的标准 PCI 热插入控制器，其中所述交换机端口允许组件的热插/拔。由 MR-PCIM 2562 使用这些热插入控制器的其中之一，用于热插/拔操作的物理方面，并被称为“物理热插入”控制器。对于每个可热插入的端口，存在这些物理热插入控制器中的一个。

另外，提供“虚拟热插入”控制器用于根联合体，其中该根联合体使用虚拟热插入控制器，以控制在交换机端口下的它们通往共享组件的逻辑连接。对于由交换机端口所支持的每个虚拟平面，存在一个虚拟热插入控制器。

对于说明性的实施方式，响应于对新 RC N 2516 的添加，在与 RC N 2516 相关联的交换机 2532 端口处，物理热插入控制器向 MR-PCIM 2562 发送“存在检测改变”中断消息，以通知 MR-PCIM 2562 新组件已被加入 PCIe 架构。此中断消息由 PCI 规范所定义，但

此处的使用是为了将中断引导至 MR-PCIM 2562 而并非引导至没有运行 MR-PCIM 2562 的根联合体。可选地，作为新组件添加的另一种通知形式，在插入前，系统管理员还可以通过通往 MR-PCIM 2562 的管理接口（未示出）来通知 MR-PCIM 2562。

MR-PCIM 2562 然后可以通过执行多个操作来处理“存在检测改变”中断，以初始化在现有 PCIe 架构 2530 中的新组件，例如，根联合体 RC N 2516。例如，MR-PCIM 2562 利用关于新添加的组件的信息来更新其架构配置数据结构 2566。在更新架构配置数据结构 2566 中，MR-PCIM 2562 执行 PCI 配置空间操作，以查询并确定该新添加组件的特征，例如，其是否为端点、根联合体、交换机等，其是何种类型的端点、根联合体、交换机等，供应商名称、零件编号、序列号等。

除了此自动查询，当所添加的组件为根联合体时，该系统管理员也可以例如通过通往 MR-PCIM 2562 的管理接口（未示出），来告知 MR-PCIM 2562 将哪些组件配置给该新添加的根联合体。例如，MR-PCIM 2562 需要知到系统管理员希望将哪些端点分配给新 RC N 2516，使得可以将那些端点添加至正确的虚拟平面，从而该新 RC N 2516 能获得对它们的访问。通过 MR-PCIM 2562 访问 MRA 交换机 2532 的 PCI 配置空间中的 PCI 多根配置结构并且如果该端点为 MRA 端点，除了在 MRA 交换机 2532 之外还访问在该端点中的 PCI 多根配置结构，来执行这种向虚拟平面的端点分配。

在描述的例子中，所添加的新组件为根联合体 RC N 2516，例如提供 RC N 2516 的刀片。然而，该组件可以是多个不同类型的组件中的任意一个，并且因此，MR-PCIM 2562 可以基于从所添加的组件收集的特征信息来确定添加组件的类型。基于确定的添加组件的类型，MR-PCIM 2562 可以执行各种操作，以动态地向 PCIe 架构 2530 添加组件，使得其可以使用于 PCIe 架构 2530 的通信和操作。

因为在所描述的例子中，添加组件为新根联合体 RC N 2516，所以 MR-PCIM 2562 将新根联合体 RC N 2516 与虚拟平面 2515 相关联，

并且然后将系统管理员已经指定给新根联合体 RC N 2516 的端点相关联，如上文的详细描述的那样。按照这种方式，新组件可以动态添加至 PCIe 架构。

随着新根联合体 RC N 2516 已由 MR-PCIM 2562 添加至 PCIe 架构 2530，通过在架构配置数据结构 2566 中包括对于新根联合体 RC N 2516 的特征信息，将该组件的 PCI 配置空间设置为与该新根联合体相关联，并将该新根联合体与虚拟平面 2515 相关联，可在 PCIe 架构 2530 中使用新根联合体 RC N 2516。上述操作可以在 PCIe 架构 2530 继续工作的同时动态地加以执行。因此，说明性的实施方式提供一种用于向运行中的 PCIe 架构 2530 中热插入组件的机制。

应该注意，架构配置数据结构 2566 可以用于多种目的，其中该架构配置数据结构 2566 由 MR-PCIM 2562 关于该系统配置保持为当前值。例如，其可用于经由 MR-PCIM 管理接口向系统管理员显示 PCIe 架构 2530 的 I/O 配置。即，可以通过由 MR-PCIM 2562 所提供的管理接口来向给系统管理员提供将哪些端点分配给哪些根联合体、哪些端点并未分配给任何根联合体并因此可用于分配等的表示。架构配置数据结构 2566 还可以跨过系统电源周期而在诸如闪存或硬盘的非易失性存储器中持续保存，使得当该系统启动时，可以由 MR-PCIM 2562 自动恢复先前的端点到根联合体的分配，如系统管理员先前分配的那样。

利用从 I/O 通信架构 2530、端点 PCI 配置寄存器已知信息以及系统管理员经由 MR-PCIM 2562 的管理接口输入的信息，来保持架构配置数据结构 2566。架构配置数据结构 2566 标识 I/O 结构的树数据结构，并因此可以用于知道当执行移除操作时将移除哪些组件，如下文所述。

在上文的例子中，添加了根联合体。当发生这种情况时，作为正常加电操作的一部分，该根联合体的固件和软件使用正常的配置访问来探测在架构配置数据结构 2566 中的 I/O 配置信息。当端点为正在添加至运行中的 I/O 通信架构 2530 的组件时，一旦系统管理员

已通过 MR-PCIM 的管理接口向该期望的根联合体添加了该组件，则在如上文所述向期望的虚拟平面配置了该端点之后，MR-PCIM 2562 信令通知该根联合体已经通过上文所涉及的虚拟热插入控制器添加了该端点。作为结果，该根联合体从该虚拟热插入控制器接收中断，以及对将配置并开始使用该新端点的软件加以初始化。因此，可以添加端点，而无需停止当前系统操作。

另外，说明性的实施方式的机制还提供了用于从 PCIe 架构 2530 动态移除组件的功能性。类似于“添加”事件，系统管理员可以经由接口、自动检测机制等向 MR-PCIM 2562 通知“移除”事件。对于“移除”事件，该事件对 MR-PCIM 2562 标识：在由 MR-PCIM 2562 管理的虚拟层级中的哪个虚拟树数据分支上，该移除操作（即，组件的热插入移除）已发生或将要发生。

在移除操作中，由架构配置数据结构 2566 标识待移除的组件。如上文所述，此架构配置数据结构 2566 标识了 I/O 组件的树数据结构。该移除操作指向在该树数据结构中待移除的组件，并且该组件以下的所有也都将被移除。例如，通过经由 MR-PCIM 2562 管理接口向系统管理员显示的图形管理界面来指向待移除的组件。在这种情况下，MR-PCIM 2562 经由架构配置数据结构 2566 而知道所选择的组件如何涉及其他组件。

优选地，可以由与该组件的虚拟热插入控制器中的状态位来标识组件的关系。在后者的情况下，虚拟热插入控制器可以向 MR-PCIM 2562 发布中断消息，MR-PCIM 2562 然后可以从该虚拟热插入控制器的寄存器中读取状态，以查看哪些组件将被移除。在这种情况下，MR-PCIM 2562 可以扫描架构配置数据结构 2566 以查找该组件，以便发现哪些组件需要被移除。

移除操作的实例包括移除端点 EP5 2552 的操作，在此情况下，基于架构配置数据结构 2566 中的树数据结构，仅将该端点标识用于移除。在另一例子中，基于架构配置数据结构 2566 的树数据结构，MRA 交换机 3 2532 的移除将涉及 MRA 交换机 3 2532 和端点

EP3-EP6 2542-2554 的移除。

MR-PCIM 2562 通过更新其虚拟 PCIe 架构配置数据结构 2566 来处理“移除”事件，以便移除与“移除”事件相关联的组件，并且在所涉及的一个或者多个树分支中更新所述组件的 PCI 配置空间，以将它们从它们先前所占用的虚拟平面内移除。在移除之后，端点可以被返回至未使用的组件池，并可以随后由系统管理员分配给另一根联合体。具体地，如果该组件是如架构配置数据结构 2566 中的 PCI 配置信息所标识的端点，则 MR-PCIM 2562 在该端点所在的虚拟树数据层级中将该端点从虚拟平面移除。如果该组件是根联合体、或提供了根联合体，则从作为此虚拟平面的部分的所有组件中移除与该组件相关联的虚拟平面。这样，除了能够动态地向 PCIe 架构添加组件之外，说明性的实施方式的机制还提供在 PCIe 架构处于操作或运行的同时从 PCIe 架构动态地移除组件的能力。

图 26 是概括了根据一个说明性的实施方式的用于向 PCIe 架构添加组件的示例性操作的流程图。如图 26 所示，该操作开始于在 MR-PCIM 中接收到指示向 PCIe 架构添加组件的“添加”事件（步骤 2610）。如上文所述，此事件可以标识组件所添加到的交换机和交换机端口，并可以例如响应于组件添加的自动检测或响应于系统管理员命令输入来产生该事件。该“添加”事件可以作为“存在检测改变”中断的部分而接收，其中“呈现检测改变”中断例如由 PCIe 交换机的物理热插入控制器响应于检测到添加新组件而发出。

MR-PCIM 收集对于添加的组件的特征信息（步骤 2620）。此收集可以包括与该组件进行通信以取回特征信息，诸如从与组件相关联的 VPD 存储设备等中取回特征信息。另外，该收集可以包括 MR-PCIM 执行 PCI 配置空间操作以确定新添加组件的这些特征。MR-PCIM 基于收集的特征信息来确定该组件是否为交换机（步骤 2630）。如果该组件为交换机，则 MR-PCIM 收集对于附接到该交换机端口的任何组件的特征信息（步骤 2640）。MR-PCIM 基于对于该组件的特征信息确定该交换机是 MRA 交换机还是基本交换机（步骤

2650）。如果该交换机为 MRA 交换机，则 MR-PCIM 配置 MRA 交换机虚拟平面表（步骤 2660）。如果该交换机为基本交换机，则 MR-PCIM 向一个虚拟平面分配交换机上所有的端口（步骤 2670）。

如果该组件并非交换机，则 MR-PCIM 确定该组件提供了新根联合体或者端点。MR-PCIM 将该根联合体或端点与虚拟平面相关联（步骤 2680）。然后，使用对于该组件的特征信息，以基于该相关联的虚拟平面来更新与 MR-PCIM 相关联的 PCIe 架构配置数据结构（步骤 2690）。此更新可以包括，例如，更新在架构配置数据结构中保持的一个或多个虚拟层级。最终，MR-PCIM 更新 PCI 配置空间 VP 标识符（步骤 2695）。然后，操作终止。

图 27 是概括了根据一个说明性的实施方式的用于从 PCIe 架构中动态地移除组件的示例性操作的流程图。如图 27 所示，该操作开始于 MR-PCIM 接收到“移除”事件（步骤 2710）。如上文所述，可以例如响应于组件移除的自动检测或着响应于系统管理员命令输入而产生该事件。

MR-PCIM 确定该正在移除的组件是否为端点（步骤 2720）。如果该组件为端点，则将该组件从虚拟平面内移除（步骤 2730），其中在与 MR-PCIM 相关联的 PCIe 架构配置数据结构中已经将该组件分配给该虚拟平面。如果该组件并非端点，则该组件为根联合体。如果该组件为根联合体，则 MR-PCIM 将与该根联合体相关联的虚拟平面从 PCIe 架构配置数据结构中作为此虚拟平面的部分的所有组件中移除（步骤 2740）。除了从 MR-PCIM 的架构配置数据结构中移除组件之外，MR-PCIM 还向受影响的组件发出 PCI 配置操作，以更新该组件中的 VP 数量（步骤 2750）。然后，该操作终止。

因此，说明性的实施方式的机制提供了用于在多个系统镜像和根联合体来共享端点的各种功能性。这些功能性包括：配置共享存储器空间用于在根联合体和端点之间的通信中使用，动态地添加或移除根联合体以及其他组件等。这些各种机制均向系统添加了当需求随时间变化时进行扩展的能力。此外，这些各种机制增强了负载

均衡、并发维护、以及其他期望的系统能力的冗余。

请特别注意，尽管已经在全功能的数据处理系统的上下文中描述了本发明，但本领域技术人员应当理解，本发明的处理能够以指令的计算机可读介质的形式以及各种形式来分发，并且应当理解，可以等同地应用本发明，而无论实际用于执行该分发的信号承载介质的特定类型。计算机可读介质的例子包括：可记录类型介质，诸如软盘、硬盘驱动器、RAM、CD-ROM、DVD-ROM；以及传输类型介质，诸如数字和模拟通信链路、使用例如射频以及光波传输的传输形式的有线或无线链路。计算机可读介质可以采用编码格式的形式，其中针对在特定数据处理系统中的实际使用，对所述编码格式进行解码。

为了示例和说明的目的，提供本发明的说明，并且本发明的说明并非旨在穷尽性的或者将本发明限制于所公开的形式。许多修改和变型对于本领域技术人员将是明显的。选择和说明这些实施方式，以便最好地解释本发明的原理和实际应用，并使得本领域的其他技术人员理解本发明，具有各种修改的各种实施方式适用于所预期的特定使用。

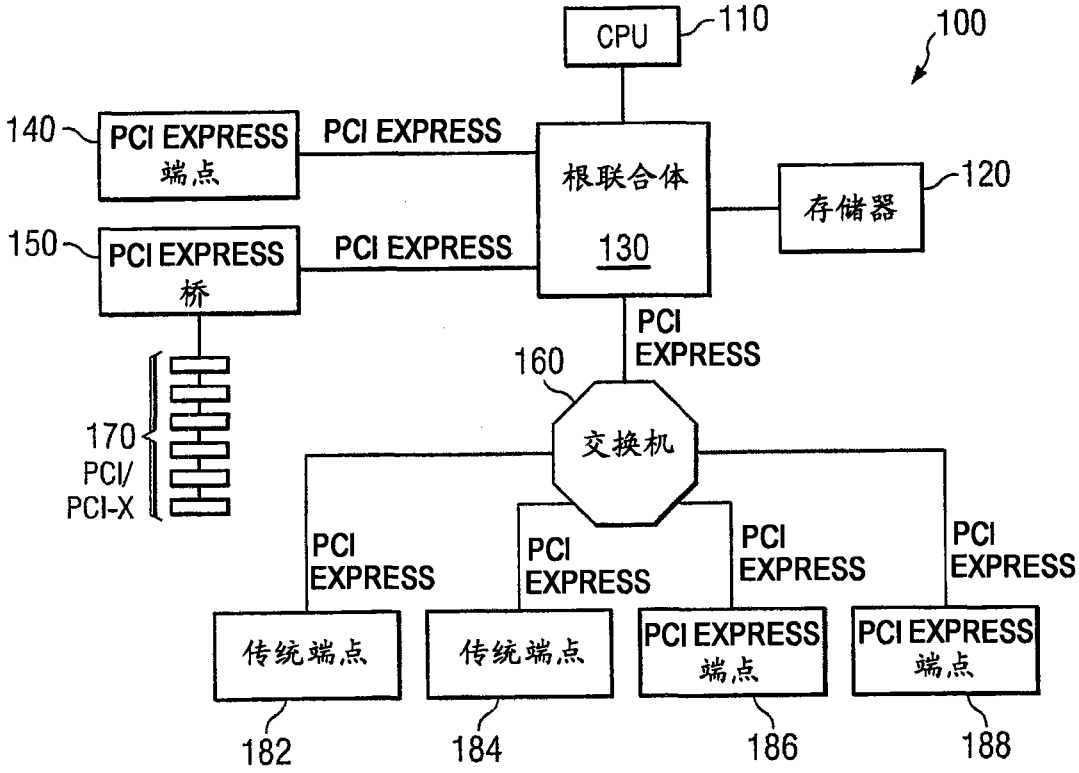

CPU
110
100
140
PCI EXPRESS
端点
PCI EXPRESS
根联合体
130
存储器
120
150
PCI EXPRESS
桥
PCI EXPRESS
PCI
EXPRESS
160
交换机
170
PCI/
PCI-X
PCI
EXPRESS
PCI
EXPRESS
PCI
EXPRESS
PCI
EXPRESS
传统端点
传统端点
PCI EXPRESS
端点
PCI EXPRESS
端点
182
184
186
188

图 1

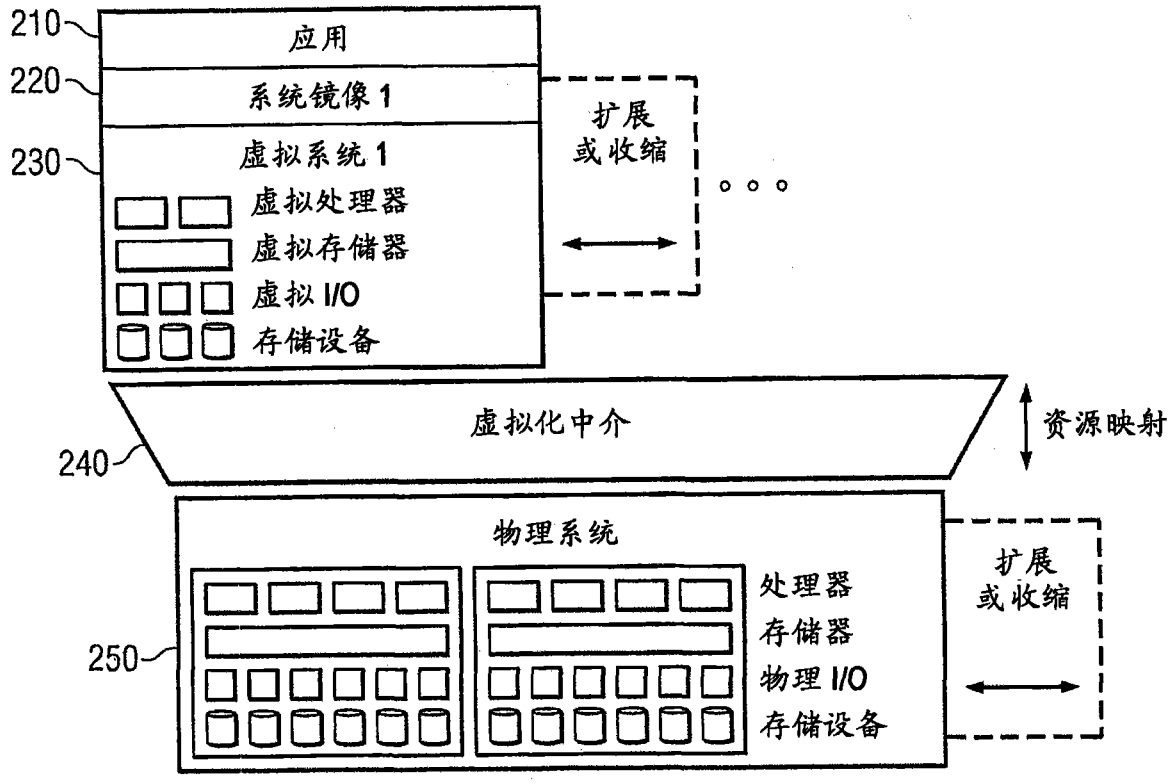

210
应用
220
系统镜像 1
230
虚拟系统 1
虚拟处理器
虚拟存储器
虚拟 I/O
存储设备
扩展
或收缩
虚拟化中介
资源映射
240
物理系统
处理器
存储器
物理 I/O
存储设备
扩展
或收缩
250

图 2

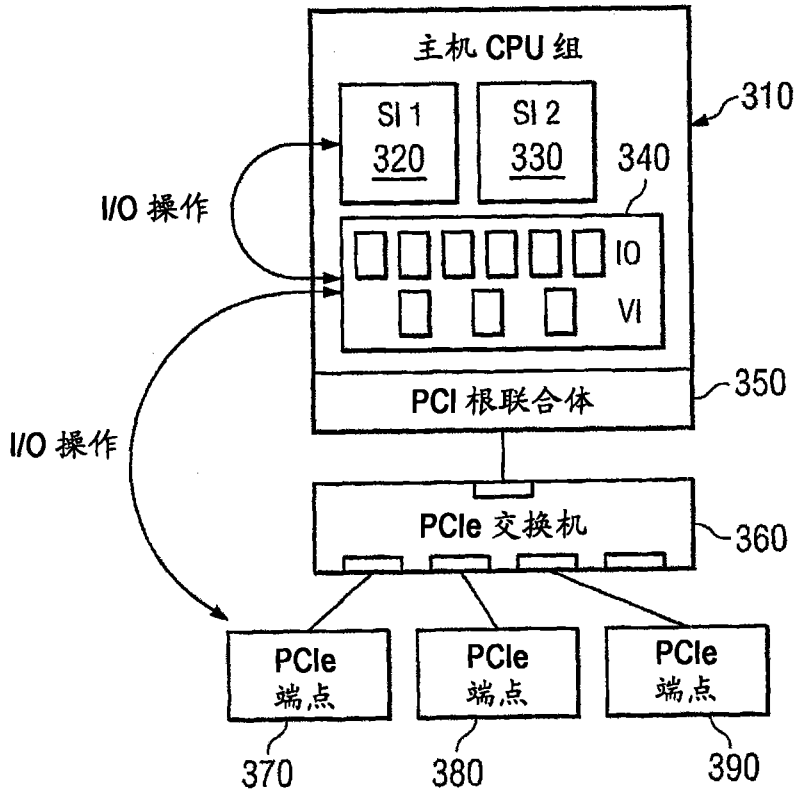

主机 CPU 组
SI 1
320
SI 2
330
340
310
I/O 操作
IO
VI
PCI 根联合体
350
I/O 操作
PCIe 交换机
360
PCIe
端点
PCIe
端点
PCIe
端点
370
380
390

图 3

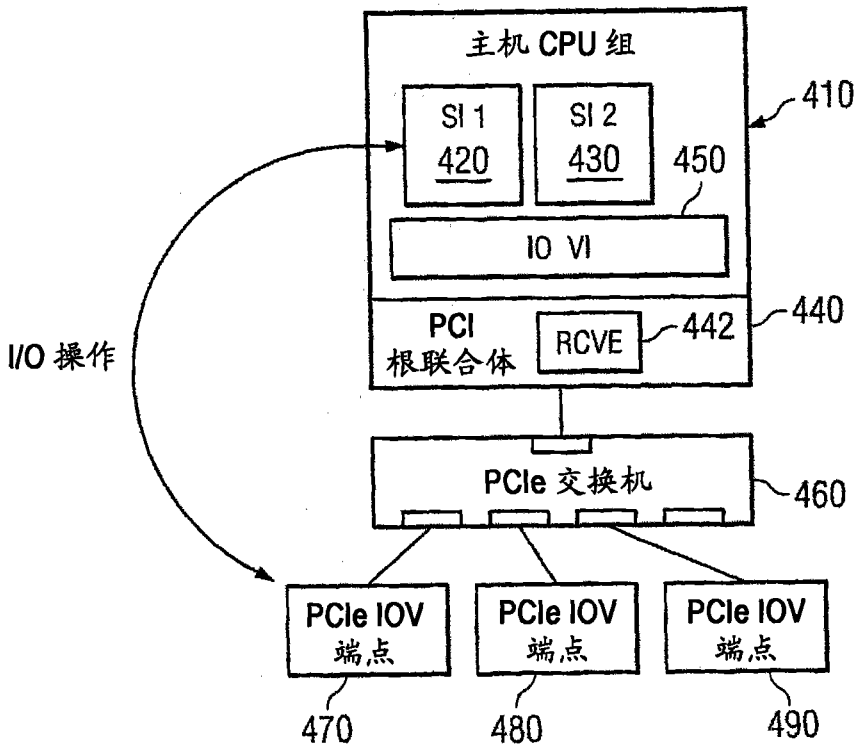

主机 CPU 组
SI 1
420
SI 2
430
410
450
IO VI
PCI
根联合体
RCVE
442
440
I/O 操作
PCIe 交换机
460
PCIe IOV
端点
PCIe IOV
端点
PCIe IOV
端点
470
480
490

图 4

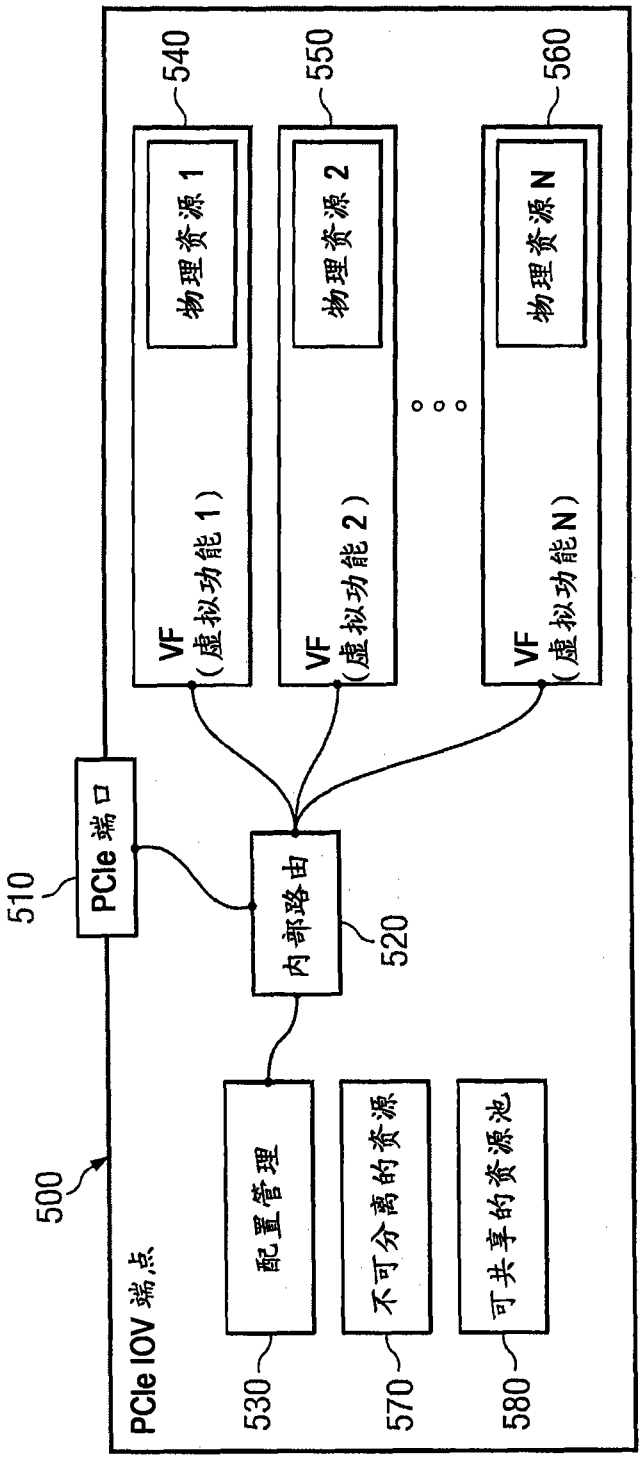

PCIe IOV 端点
500
PCIe 端口
510
内部路由
520
VF
(虚拟功能 1)
物理资源 1
540
VF
(虚拟功能 2)
物理资源 2
550
VF
(虚拟功能 N)
物理资源 N
560
配置管理
530
不可分离的资源
570
可共享的资源池
580

图 5

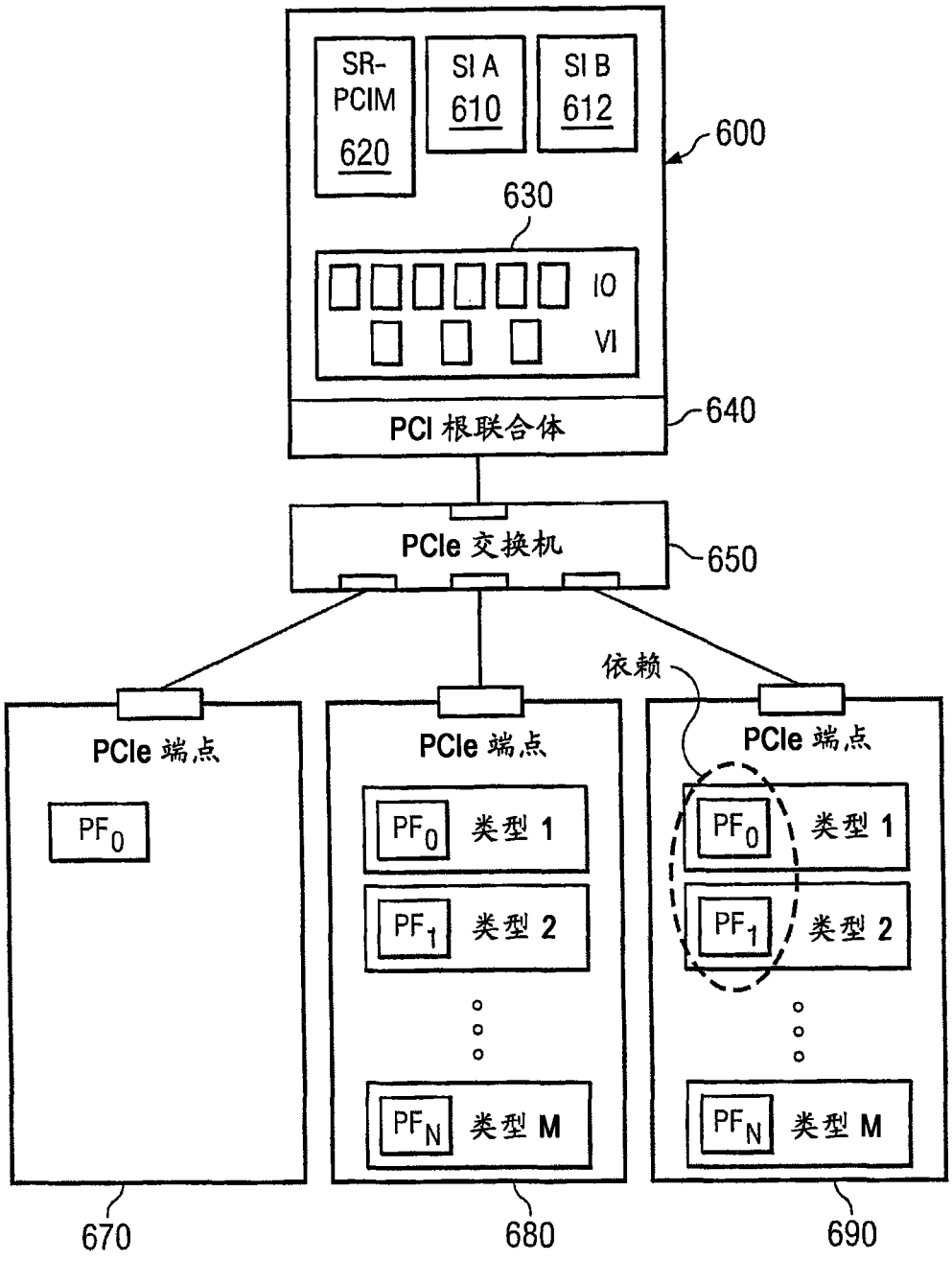

SR-PCIM 620
SI A 610
SI B 612
600
630
IO
VI
PCI 根联合体
640
PCIe 交换机
650
依赖
PCIe 端点
PF0
PCIe 端点
PF0 类型 1
PF1 类型 2
PFN 类型 M
PCIe 端点
PF0 类型 1
PF1 类型 2
PFN 类型 M
670
680
690

图 6

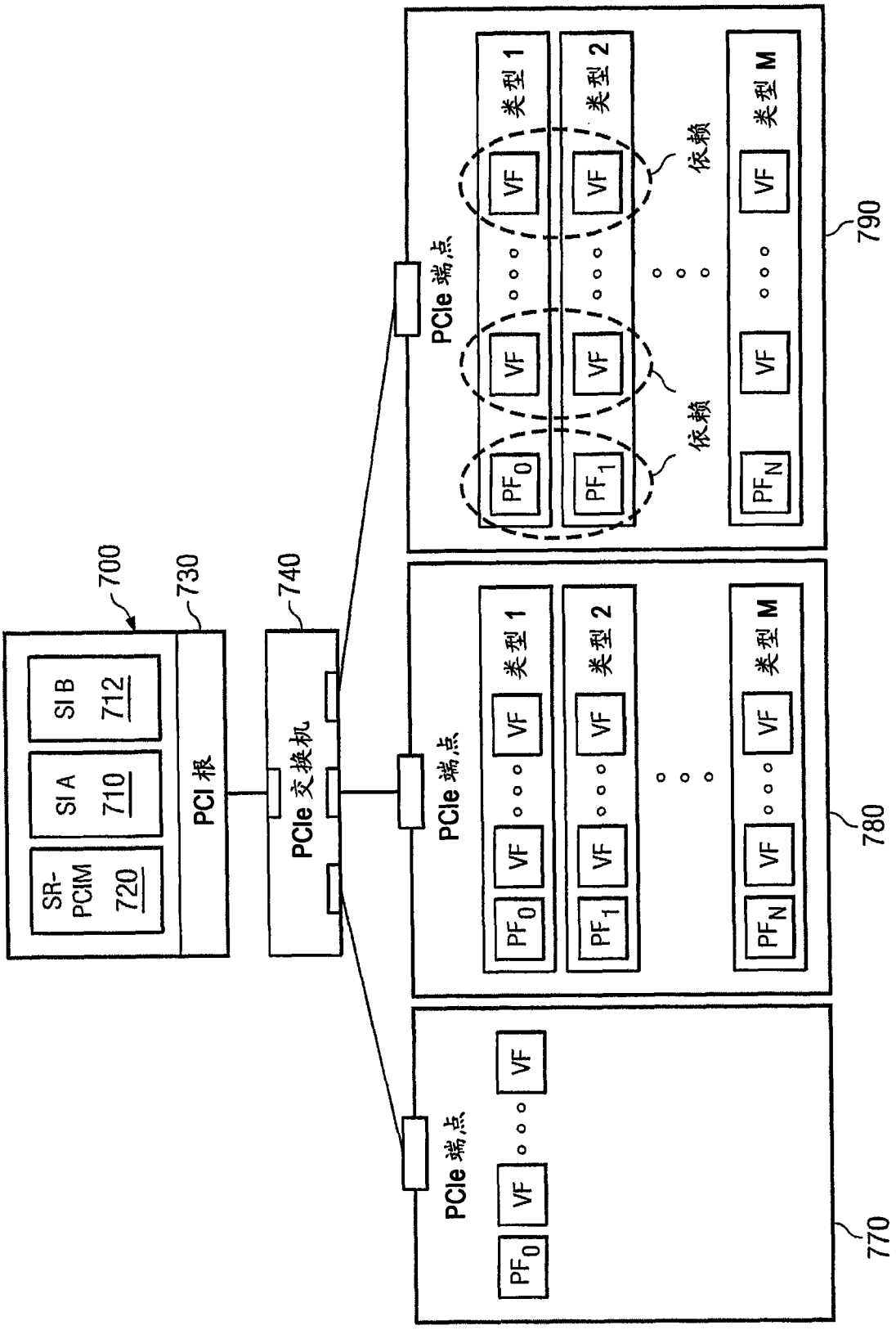

700
730
740
SR-PCIM 720
SI A 710
SI B 712
PCI 根
PCIe 交换机
PCIe 端点
PF0
VF
类型 1
类型 2
类型 M
PF1
PFN
依赖
770
780
790

图 7

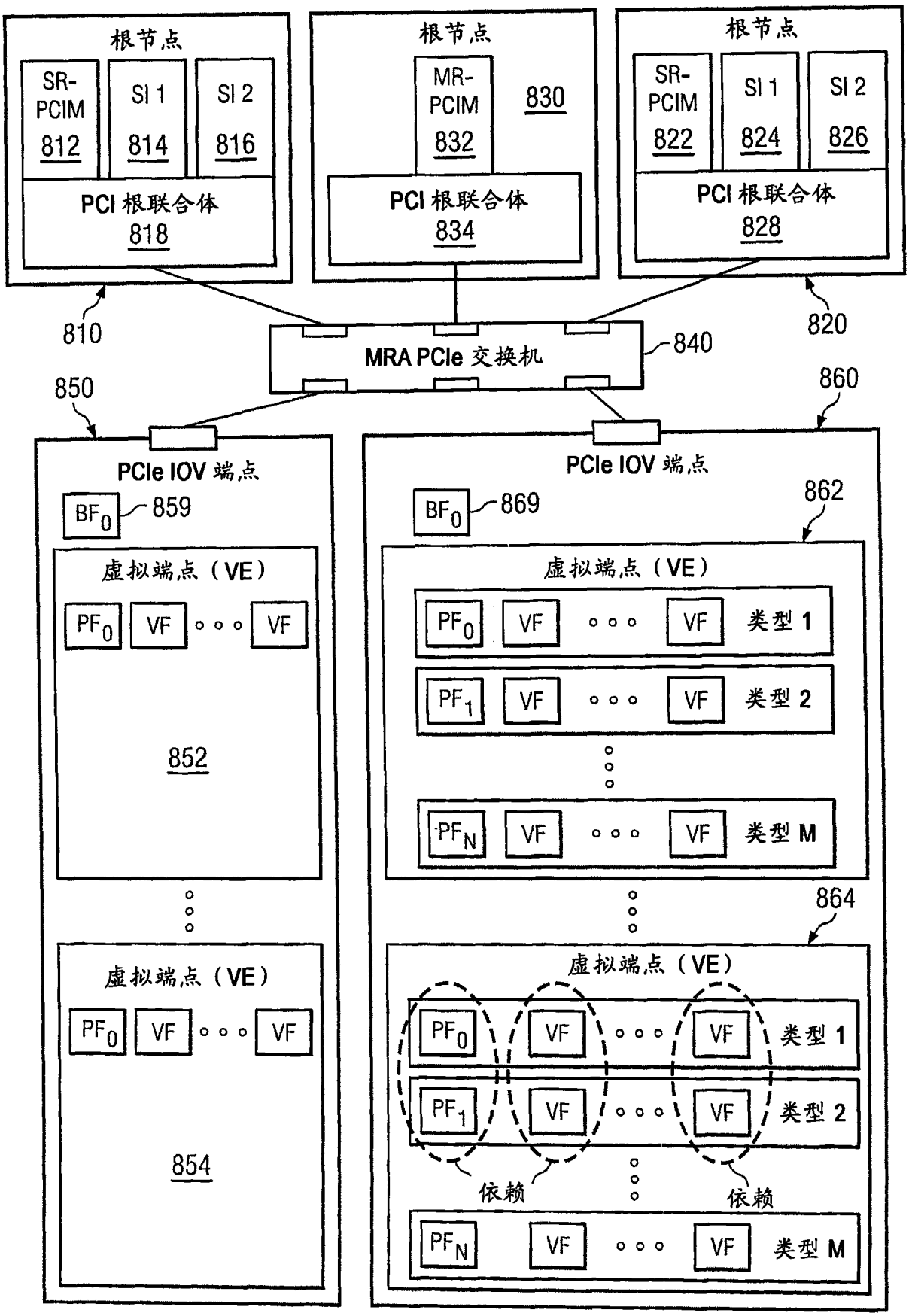

根节点
SR-PCIM 812
SI 1 814
SI 2 816
PCI 根联合体 818
根节点
MR-PCIM 832
830
PCI 根联合体 834
根节点
SR-PCIM 822
SI 1 824
SI 2 826
PCI 根联合体 828
810
820
MRA PCIe 交换机
840
850
860
PCIe IOV 端点
BF0 859
虚拟端点（VE）
PF0 VF ... VF
852
虚拟端点（VE）
PF0 VF ... VF
854
PCIe IOV 端点
BF0 869
862
虚拟端点（VE）
PF0 VF ... VF 类型 1
PF1 VF ... VF 类型 2
PFN VF ... VF 类型 M
864
虚拟端点（VE）
PF0 VF ... VF 类型 1
PF1 VF ... VF 类型 2
依赖
依赖
PFN VF ... VF 类型 M

图 8

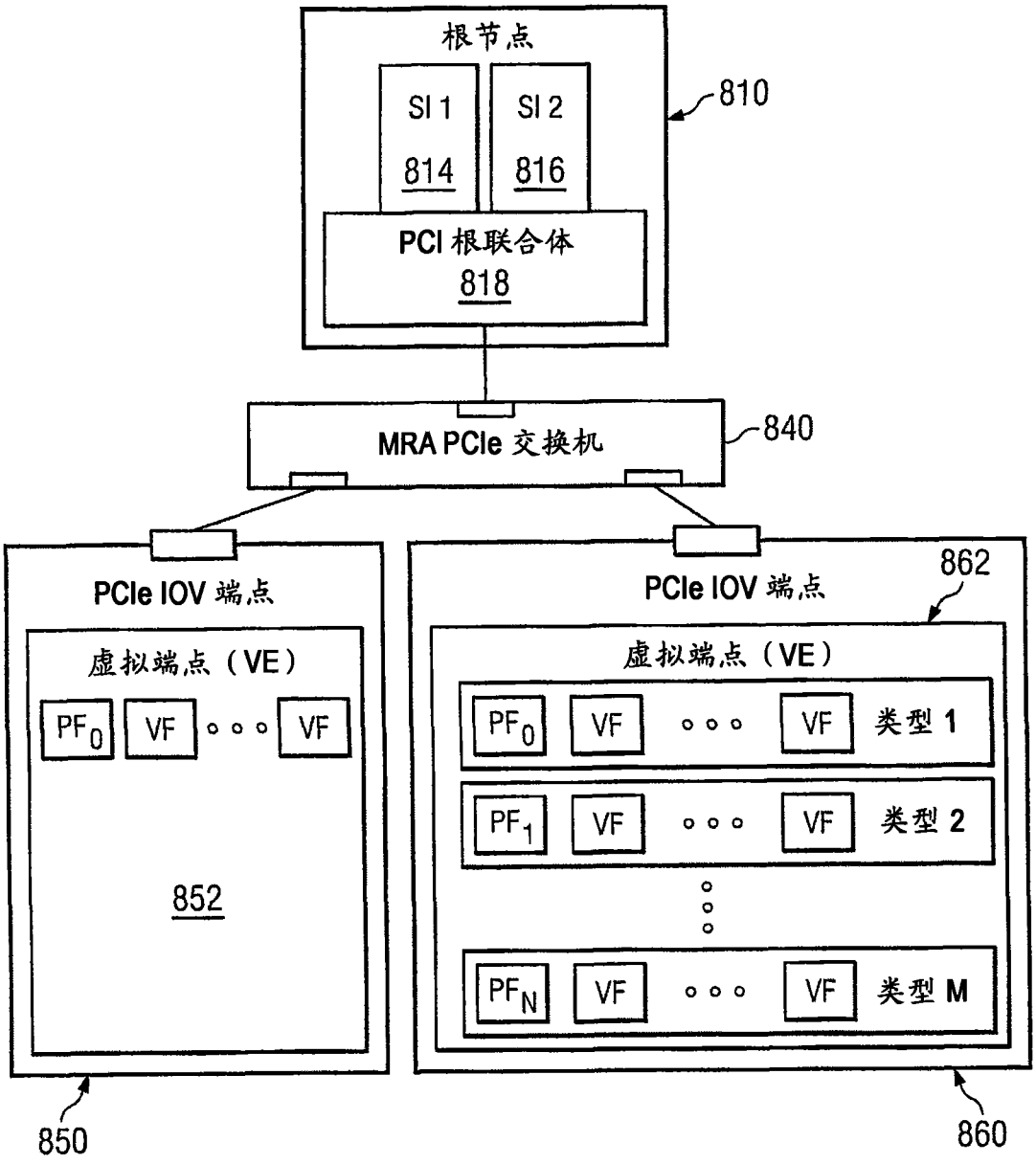

根节点
SI 1
814
SI 2
816
810
PCI 根联合体
818
MRA PCIe 交换机
840
PCIe IOV 端点
虚拟端点（VE）
PF0
VF
VF
852
850
PCIe IOV 端点
862
虚拟端点（VE）
PF0
VF
VF
类型 1
PF1
VF
VF
类型 2
PFN
VF
VF
类型 M
860

图 9

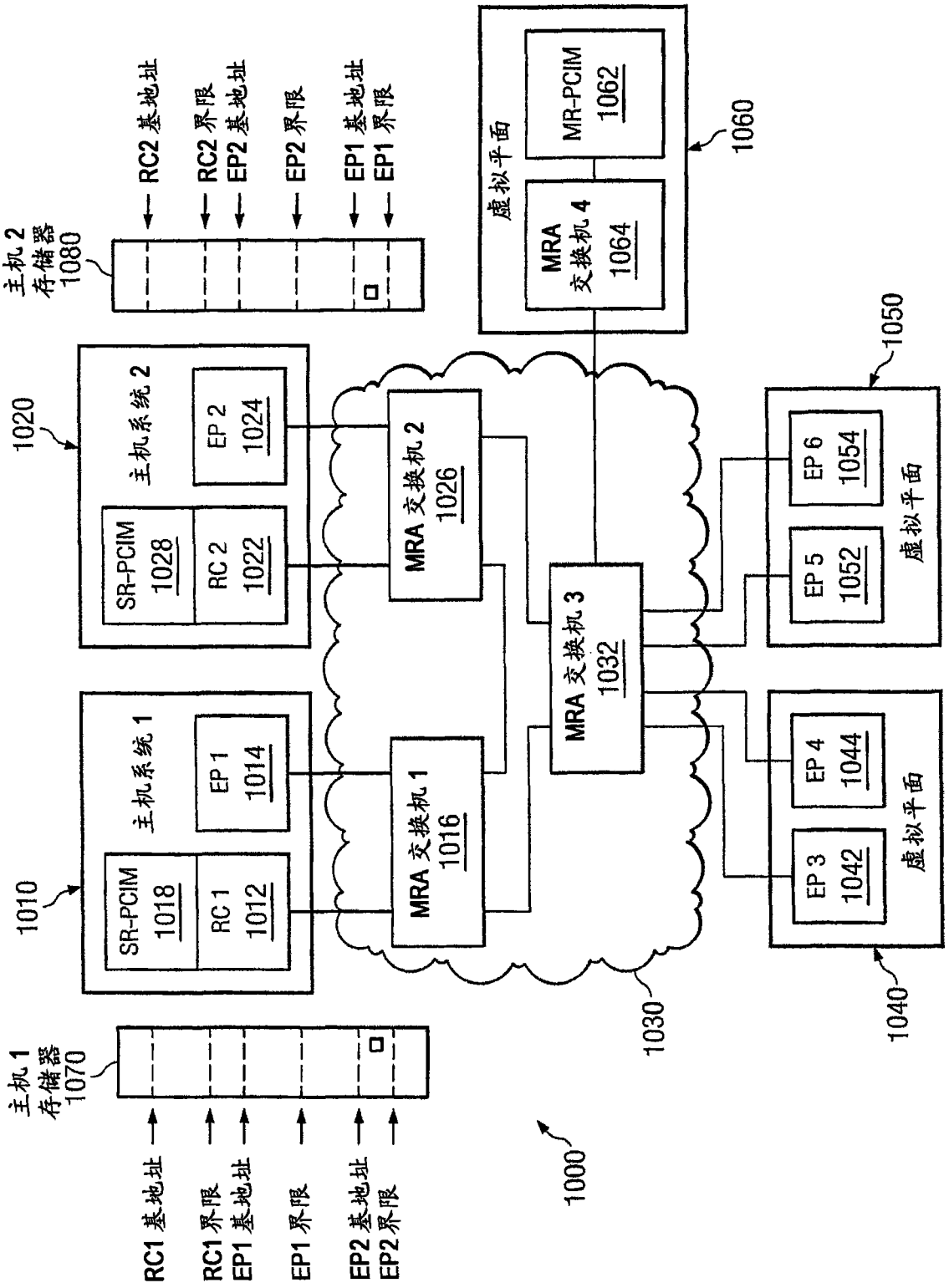

主机 1
存储器
1070
RC1 基地址
RC1 界限
EP1 基地址
EP1 界限
EP2 基地址
EP2 界限
1000
1010
主机系统 1
SR-PCIM
1018
RC 1
1012
EP 1
1014
1020
主机系统 2
SR-PCIM
1028
RC 2
1022
EP 2
1024
主机 2
存储器
1080
RC2 基地址
RC2 界限
EP2 基地址
EP2 界限
EP1 基地址
EP1 界限
MRA 交换机 1
1016
MRA 交换机 2
1026
MRA 交换机 3
1032
1030
1040
EP 3
1042
EP 4
1044
虚拟平面
1050
EP 5
1052
EP 6
1054
虚拟平面
1060
虚拟平面
MRA
交换机 4
1064
MR-PCIM
1062

图 10

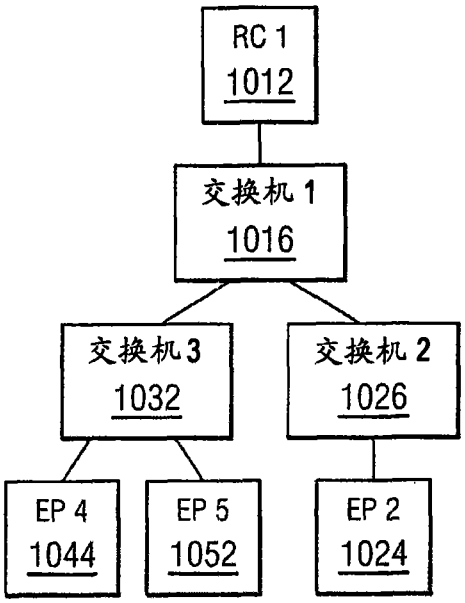


图 11A

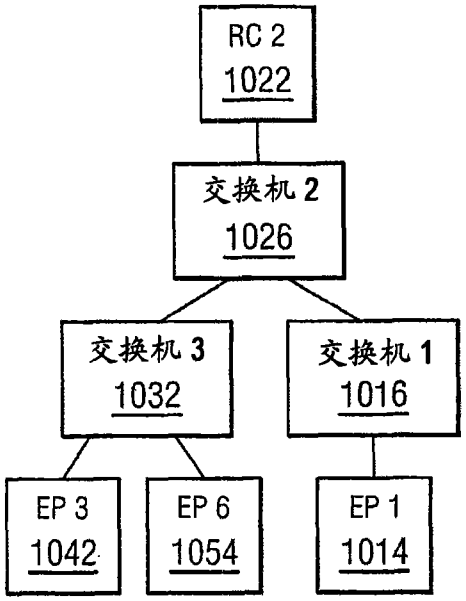


图 11B

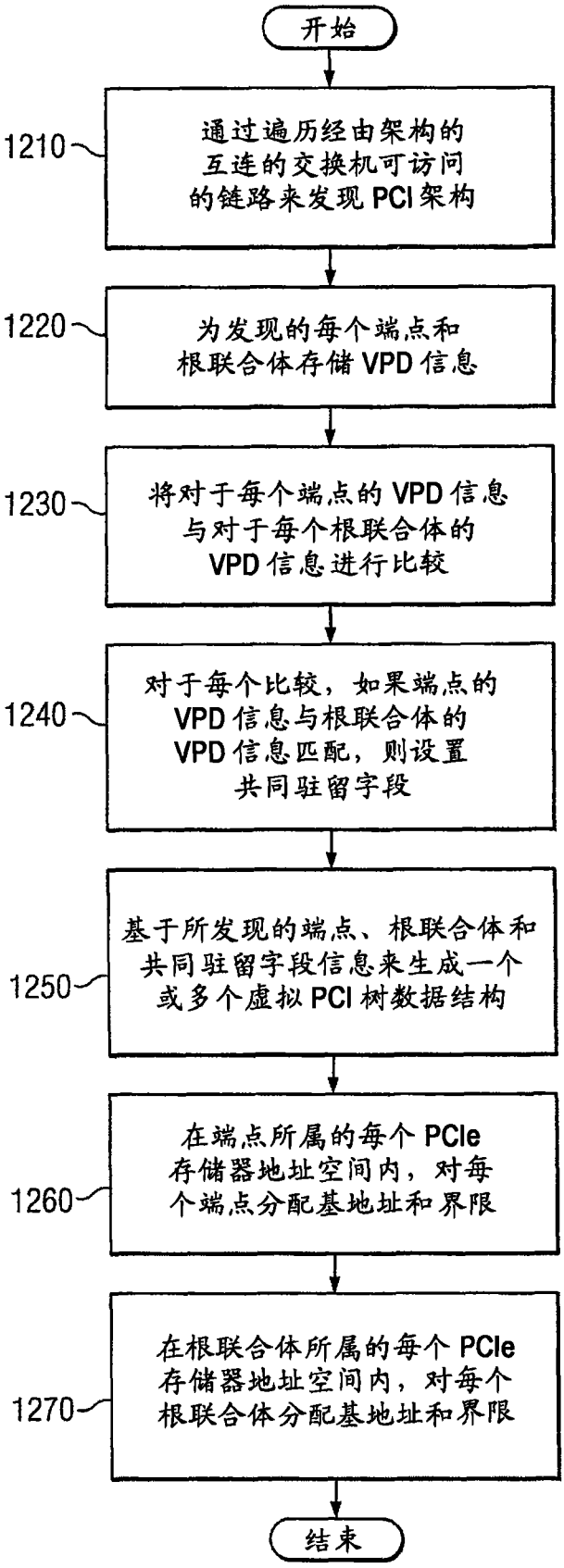


图 12

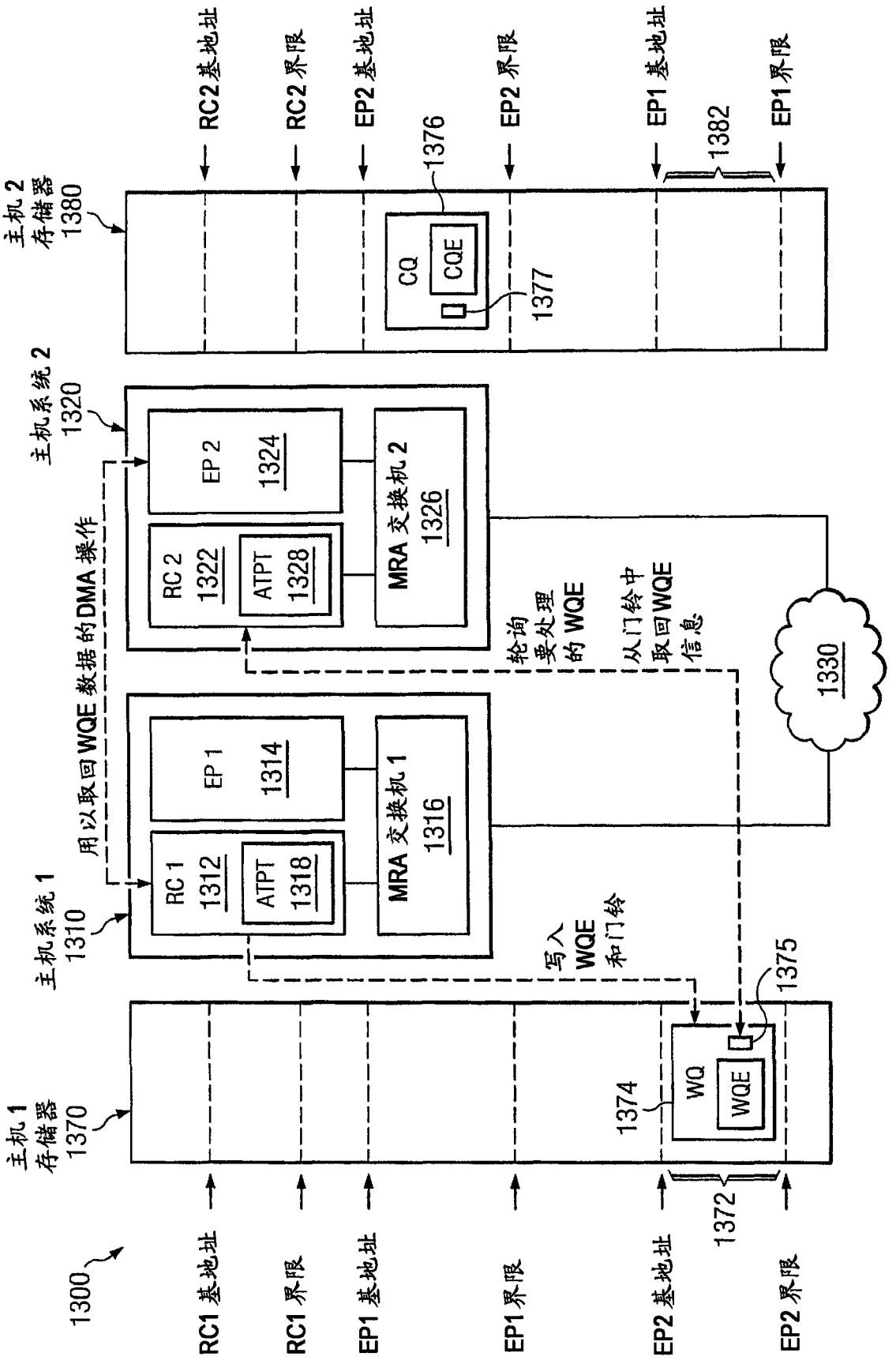

1300
主机 1 存储器 1370
主机系统 1 1310
主机系统 2 1320
主机 2 存储器 1380
RC1 基地址
RC1 界限
EP1 基地址
EP1 界限
EP2 基地址
EP2 界限
RC2 基地址
RC2 界限
EP2 基地址
EP2 界限
EP1 基地址
EP1 界限
1372
1374
1375
1376
1377
1382
RC 1 1312
ATPT 1318
EP 1 1314
MRA 交换机 1 1316
RC 2 1322
ATPT 1328
EP 2 1324
MRA 交换机 2 1326
WQ
WQE
CQ
CQE
1330
用以取回 WQE 数据的 DMA 操作
写入 WQE 和门铃
轮询要处理的 WQE
从门铃中取回 WQE 信息

图 13

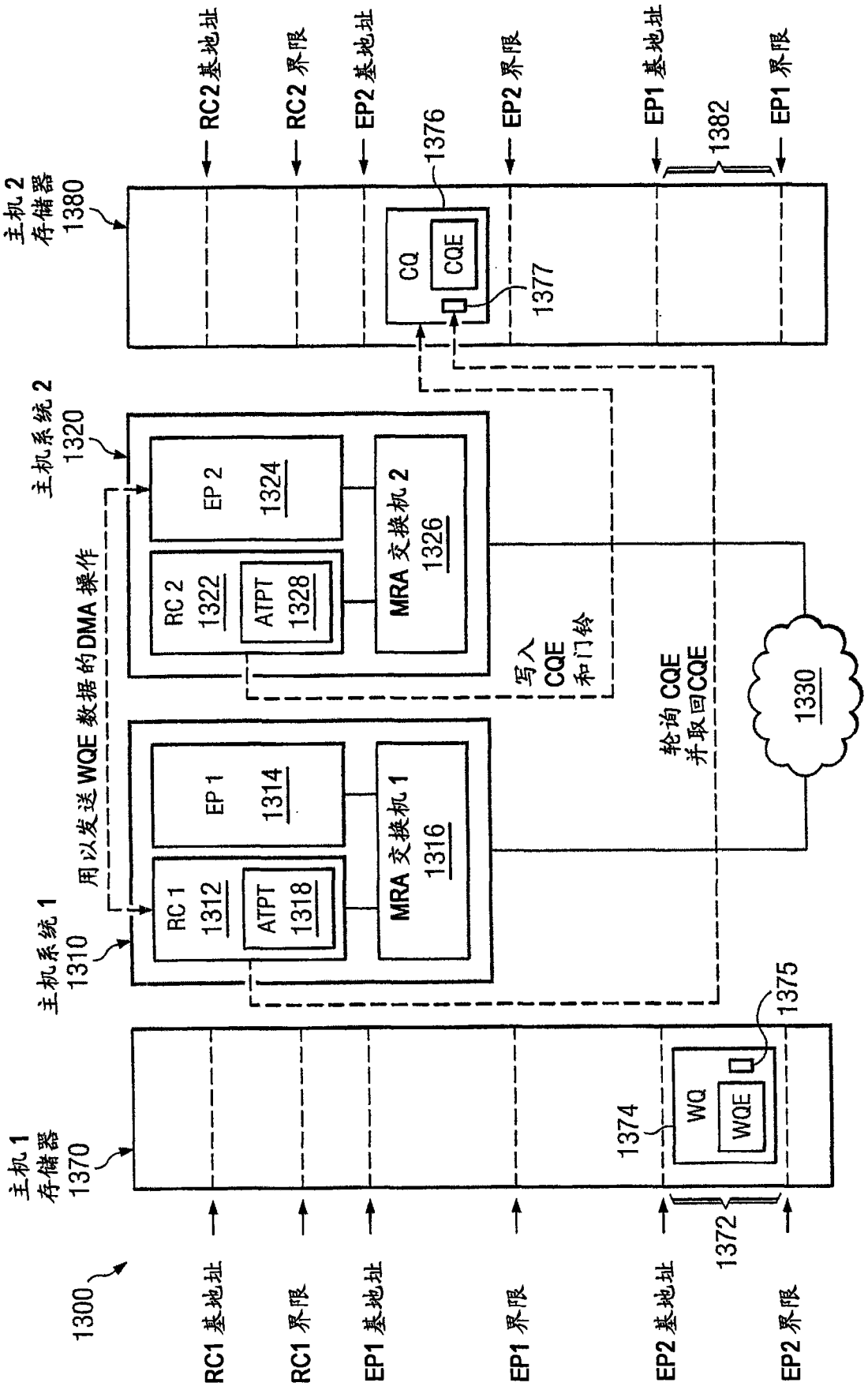

1300
主机 1 存储器 1370
RC1 基地址
RC1 界限
EP1 基地址
EP1 界限
EP2 基地址
EP2 界限
1372
1374
WQ
WQE
1375
主机系统 1 1310
RC 1 1312
ATPT 1318
EP 1 1314
MRA 交换机 1 1316
用以发送 WQE 数据的DMA 操作
主机系统 2 1320
RC 2 1322
ATPT 1328
EP 2 1324
MRA 交换机 2 1326
写入 CQE 和门铃
轮询 CQE 并取回CQE
1330
主机 2 存储器 1380
RC2 基地址
RC2 界限
EP2 基地址
1376
EP2 界限
CQ
CQE
1377
EP1 基地址
1382
EP1 界限

图 14

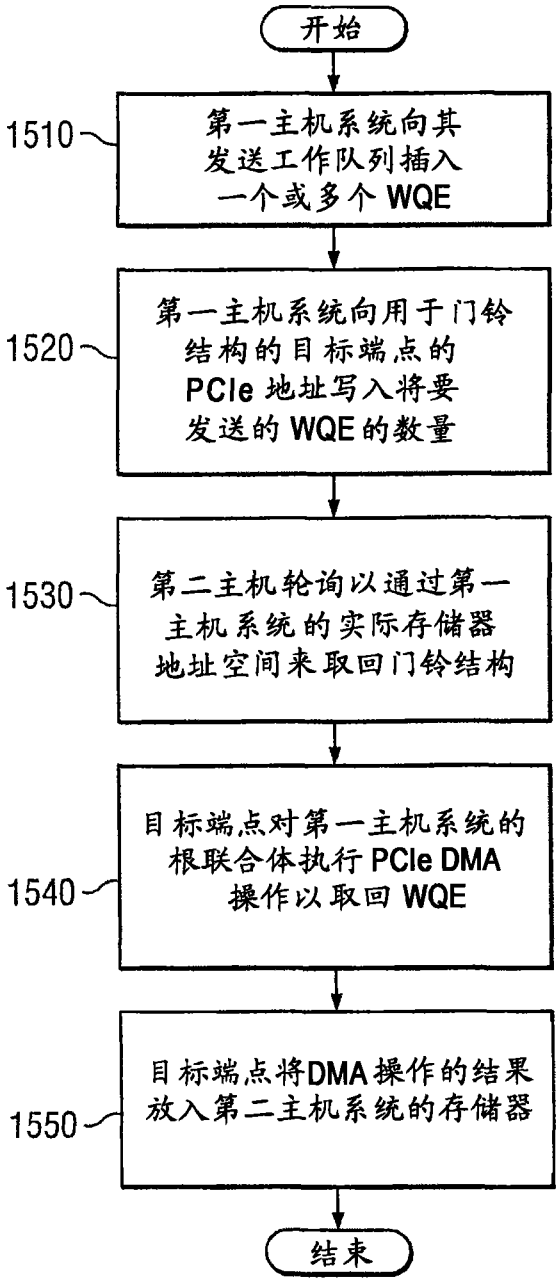

开始
1510
第一主机系统向其发送工作队列插入一个或多个 WQE
1520
第一主机系统向用于门铃结构的目标端点的 PCIe 地址写入将要发送的 WQE 的数量
1530
第二主机轮询以通过第一主机系统的实际存储器地址空间来取回门铃结构
1540
目标端点对第一主机系统的根联合体执行 PCIe DMA 操作以取回 WQE
1550
目标端点将DMA操作的结果放入第二主机系统的存储器
结束

图 15

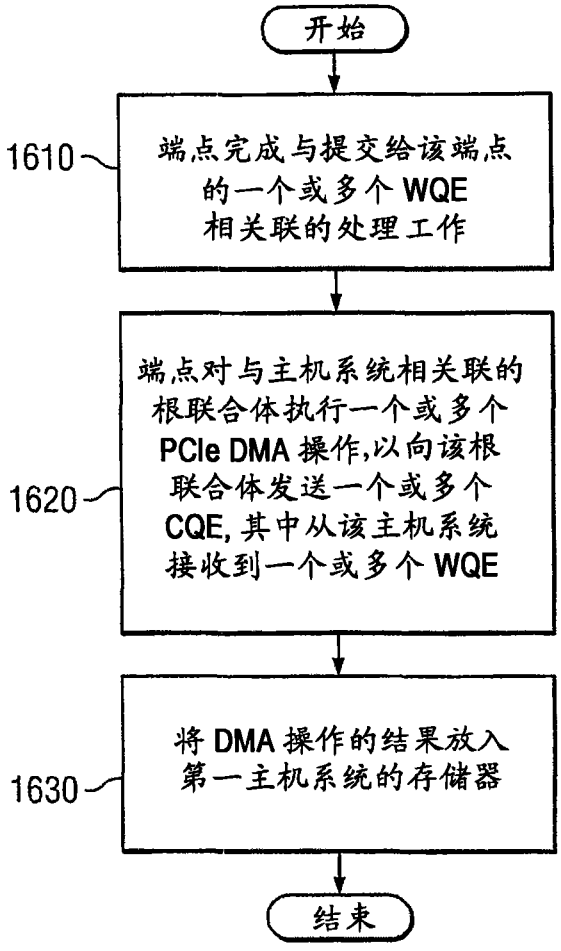

开始
1610
端点完成与提交给该端点的一个或多个 WQE 相关联的处理工作
1620
端点对与主机系统相关联的根联合体执行一个或多个 PCIe DMA 操作,以向该根联合体发送一个或多个 CQE,其中从该主机系统接收到一个或多个 WQE
1630
将 DMA 操作的结果放入第一主机系统的存储器
结束

图 16

命令	数据	响应
推送	推送	推送
推送	推送	拉回
推送	拉回	推送
推送	拉回	拉回
拉回	推送	推送
拉回	推送	拉回
拉回	拉回	推送
拉回	拉回	拉回

图 17

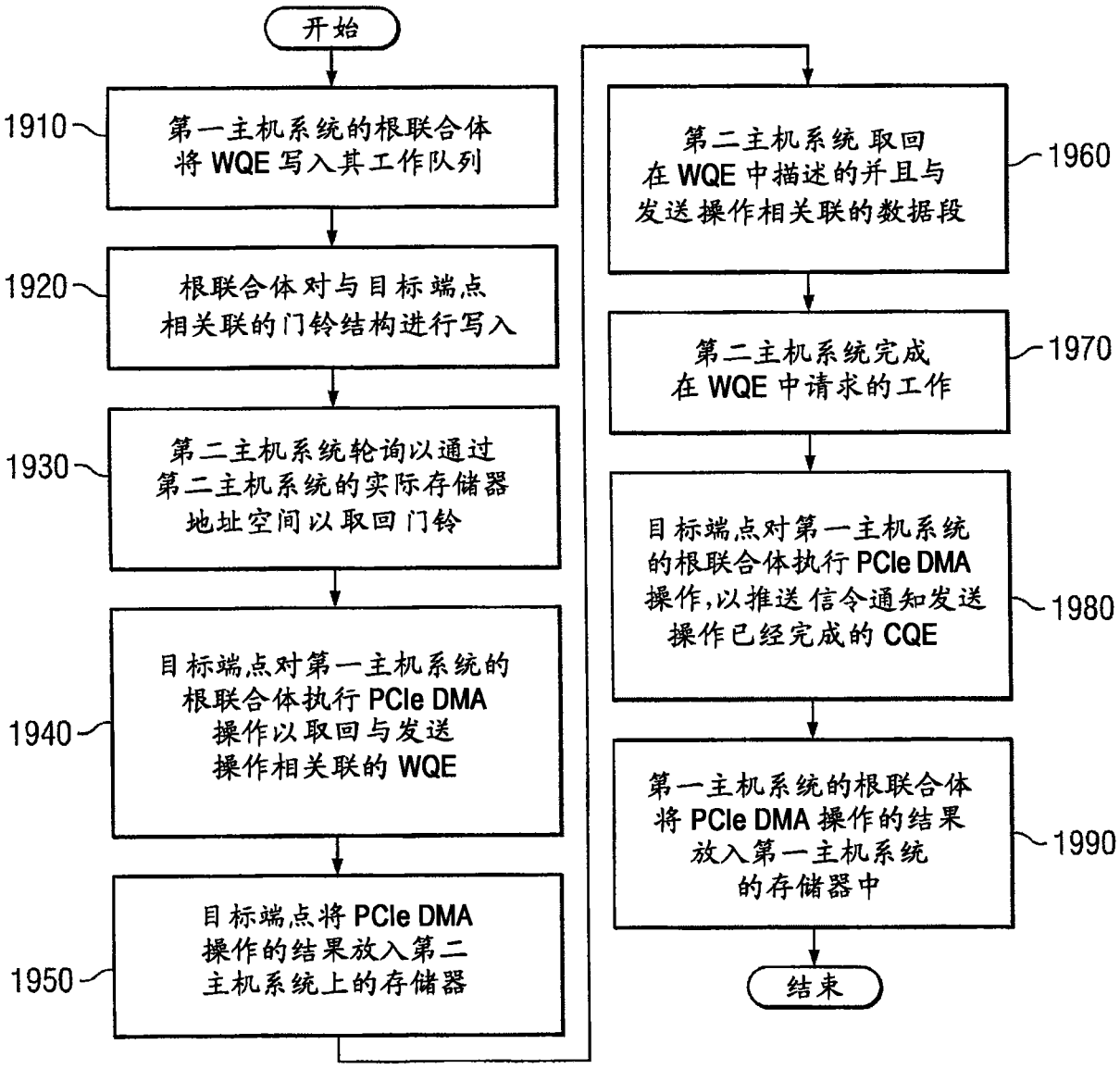


图 19

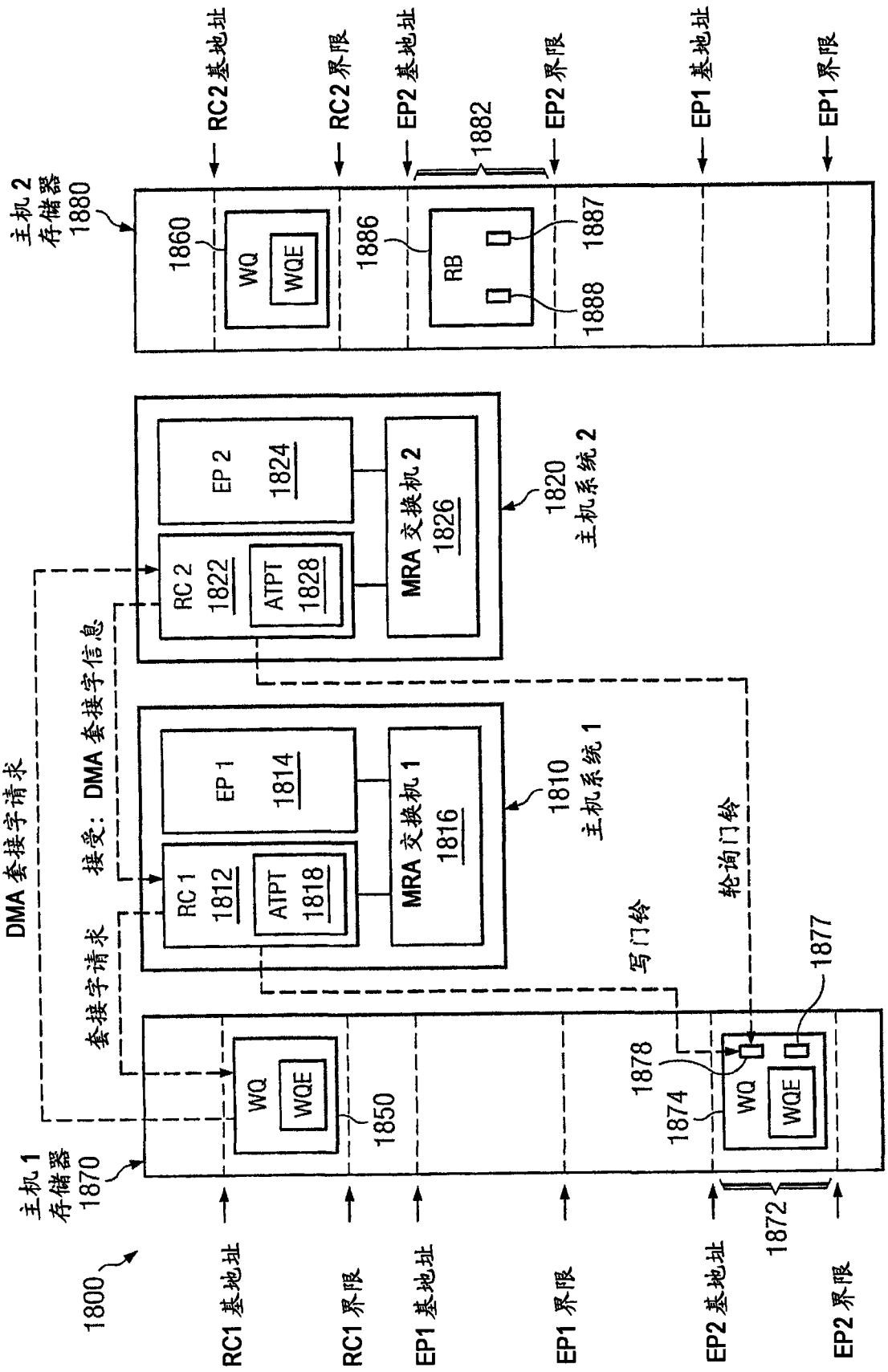

主机 2
存储器
1880
RC2 基地址
RC2 界限
EP2 基地址
EP2 界限
EP1 基地址
EP1 界限
1882
1860
WQ
WQE
1886
RB
1887
1888
1820
主机系统 2
EP 2
1824
RC 2
1822
ATPT
1828
MRA 交换机 2
1826
DMA 套接字请求
套接字请求
接受: DMA 套接字信息
1810
主机系统 1
EP 1
1814
RC 1
1812
ATPT
1818
MRA 交换机 1
1816
写门铃
轮询门铃
1877
1878
1874
1850
主机 1
存储器
1870
1800
1872

图 18

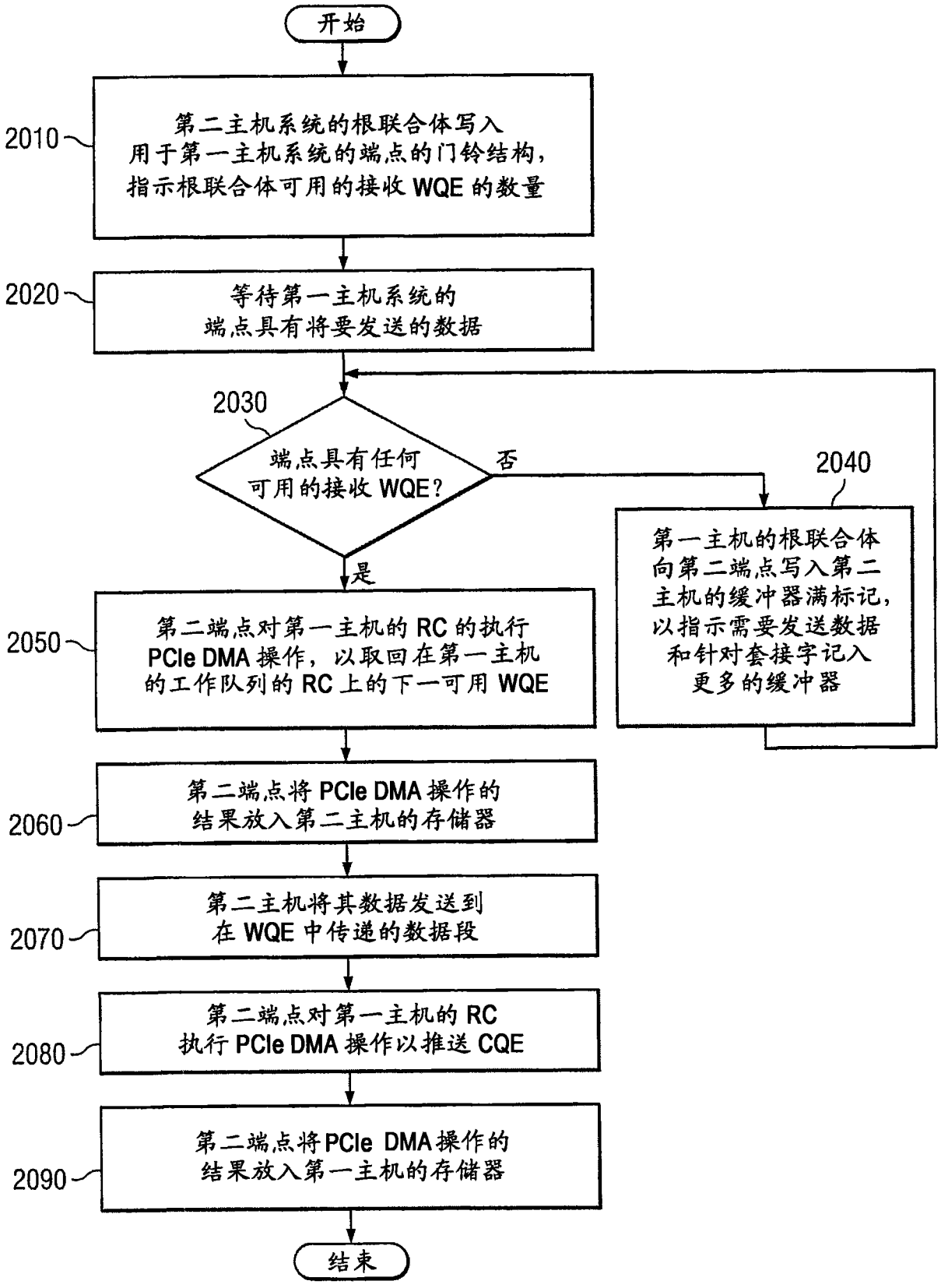

开始
2010
第二主机系统的根联合体写入用于第一主机系统的端点的门铃结构，指示根联合体可用的接收 WQE 的数量
2020
等待第一主机系统的端点具有将要发送的数据
2030
端点具有任何可用的接收 WQE?
否
2040
第一主机的根联合体向第二端点写入第二主机的缓冲器满标记，以指示需要发送数据和针对套接字记入更多的缓冲器
是
2050
第二端点对第一主机的 RC 的执行 PCIe DMA 操作，以取回在第一主机的工作队列的 RC 上的下一可用 WQE
2060
第二端点将 PCIe DMA 操作的结果放入第二主机的存储器
2070
第二主机将其数据发送到在 WQE 中传递的数据段
2080
第二端点对第一主机的 RC 执行 PCIe DMA 操作以推送 CQE
2090
第二端点将 PCIe DMA 操作的结果放入第一主机的存储器
结束

图 20

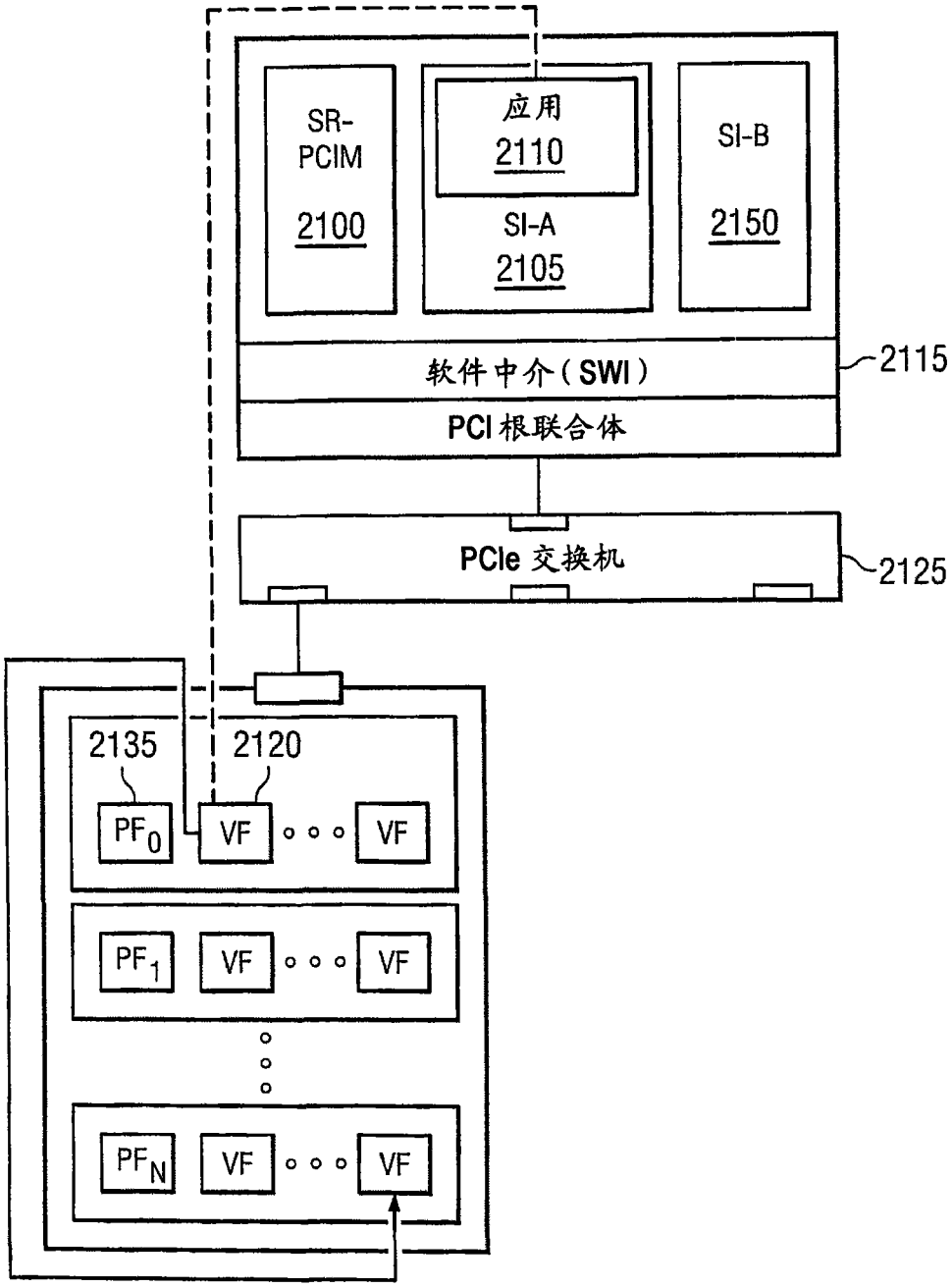

SR-
PCIM
2100
应用
2110
SI-A
2105
SI-B
2150
软件中介(SWI)
PCI根联合体
2115
PCIe 交换机
2125
2135
2120
PF0
VF
VF
PF1
VF
VF
PFN
VF
VF

图 21A

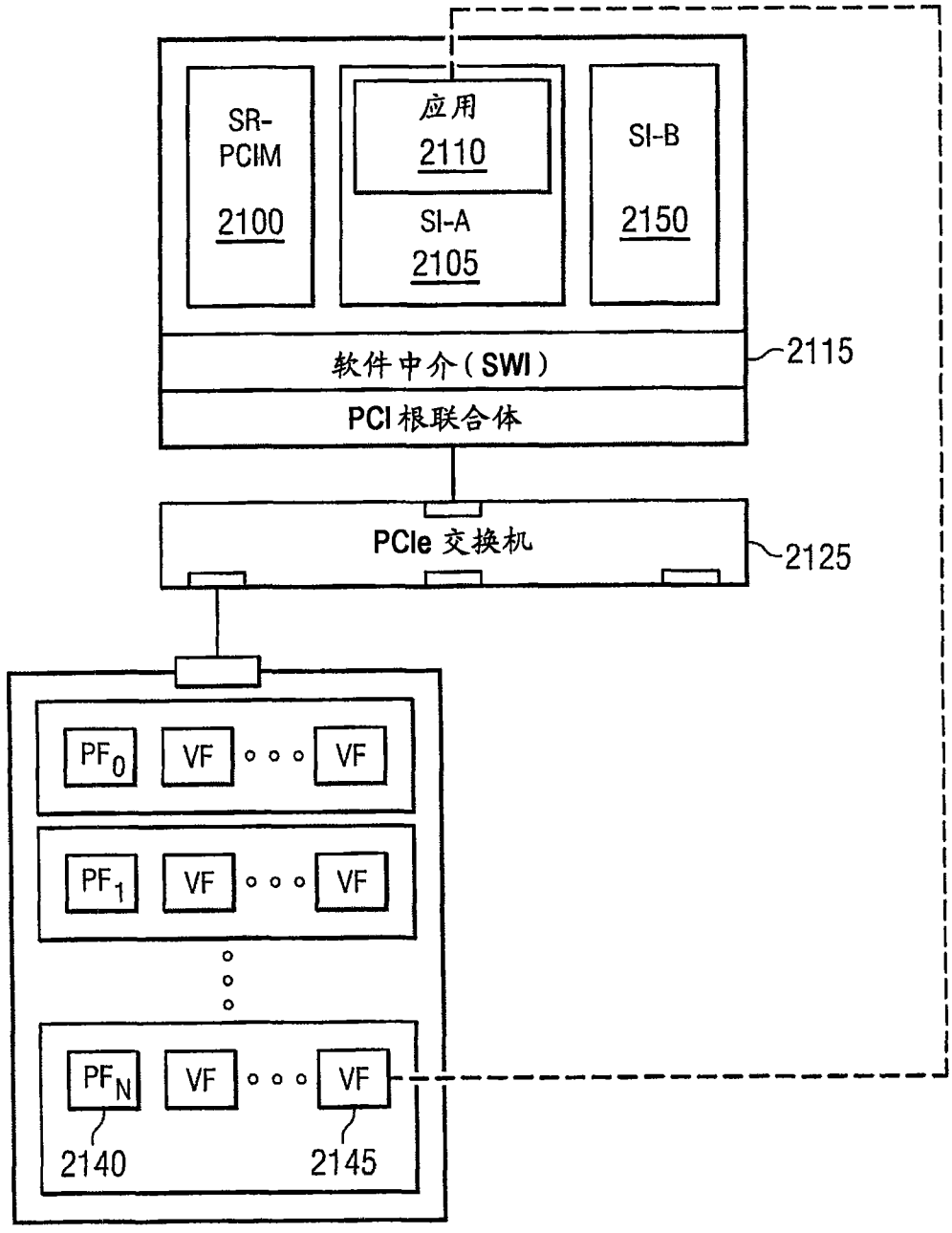

SR-
PCIM
2100
应用
2110
SI-A
2105
SI-B
2150
软件中介(SWI)
2115
PCI根联合体
PCIe 交换机
2125
PF0
VF
VF
PF1
VF
VF
PFN
VF
VF
2140
2145

图 21B

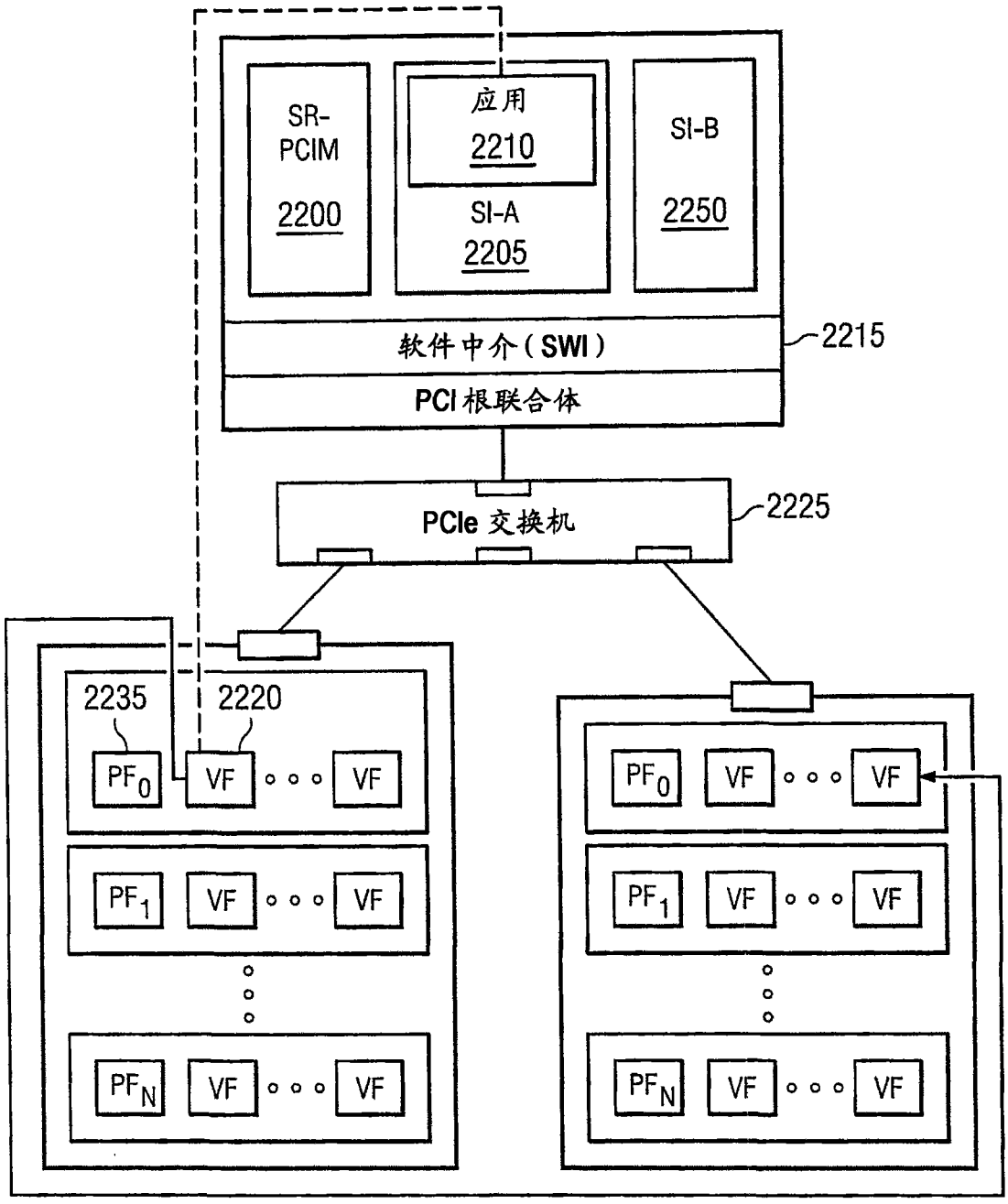

SR-PCIM
2200
应用
2210
SI-A
2205
SI-B
2250
软件中介(SWI)
2215
PCI根联合体
PCIe 交换机
2225
2235
2220
PF0
VF
PF1
PFN

图 22A

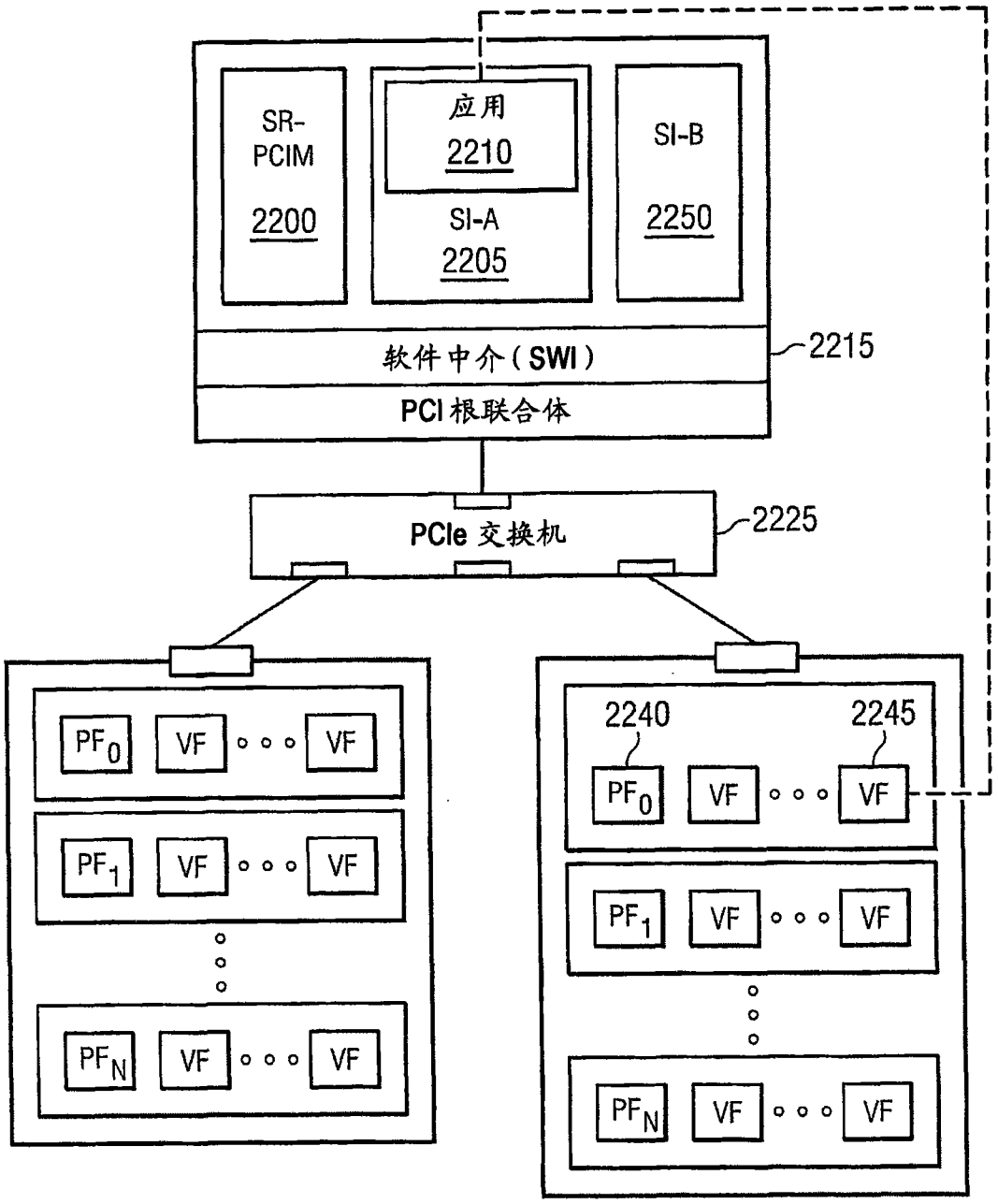

SR-PCIM
2200
应用
2210
SI-A
2205
SI-B
2250
软件中介(SWI)
2215
PCI根联合体
PCIe 交换机
2225
PF0
VF
VF
PF1
VF
VF
PFN
VF
VF
2240
2245
PF0
VF
VF
PF1
VF
VF
PFN
VF
VF

图 22B

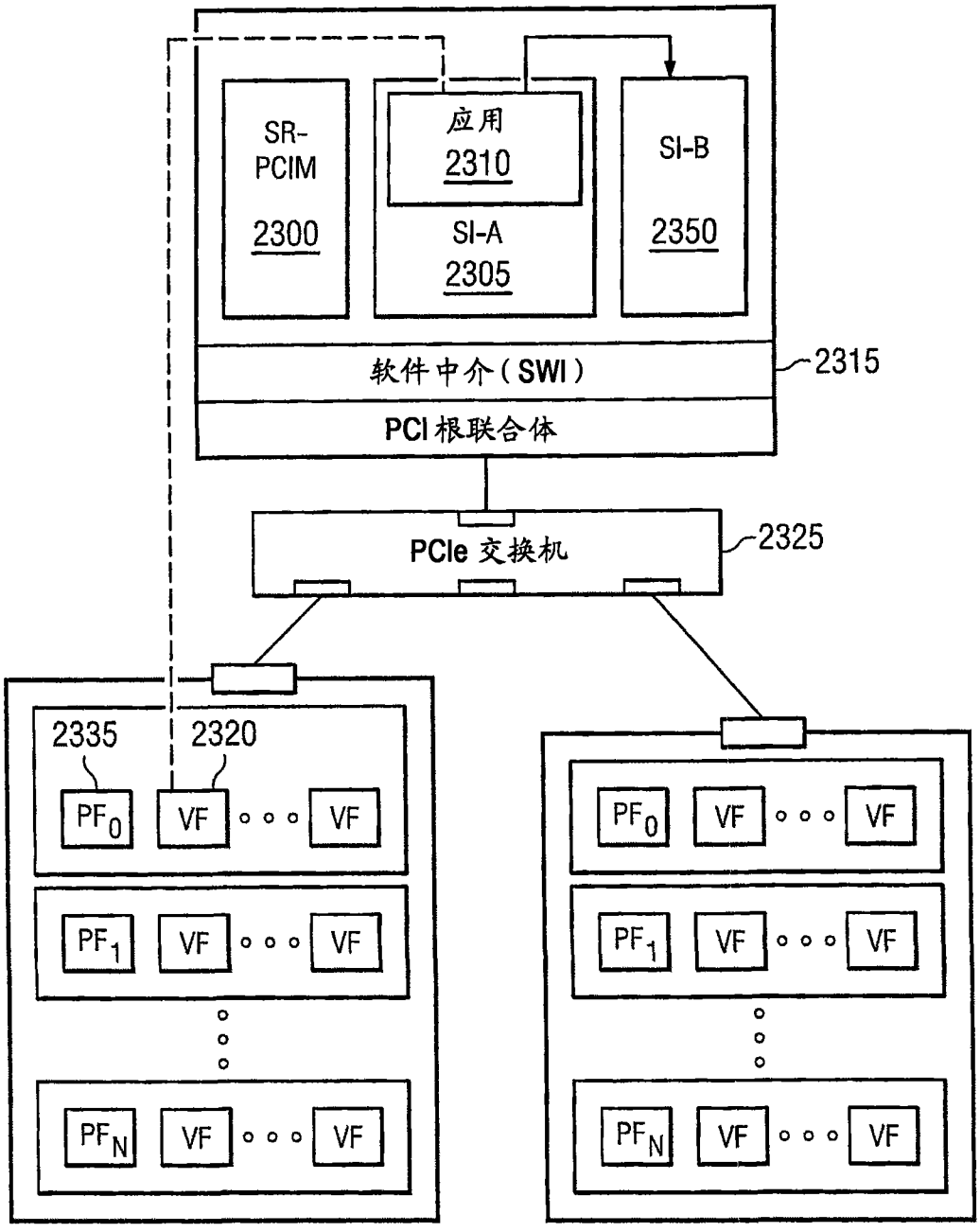

SR-
PCIM
2300
应用
2310
SI-A
2305
SI-B
2350
软件中介(SWI)
2315
PCI根联合体
PCIe 交换机
2325
2335
2320
PF0
VF
PF1
PFN

图 23A

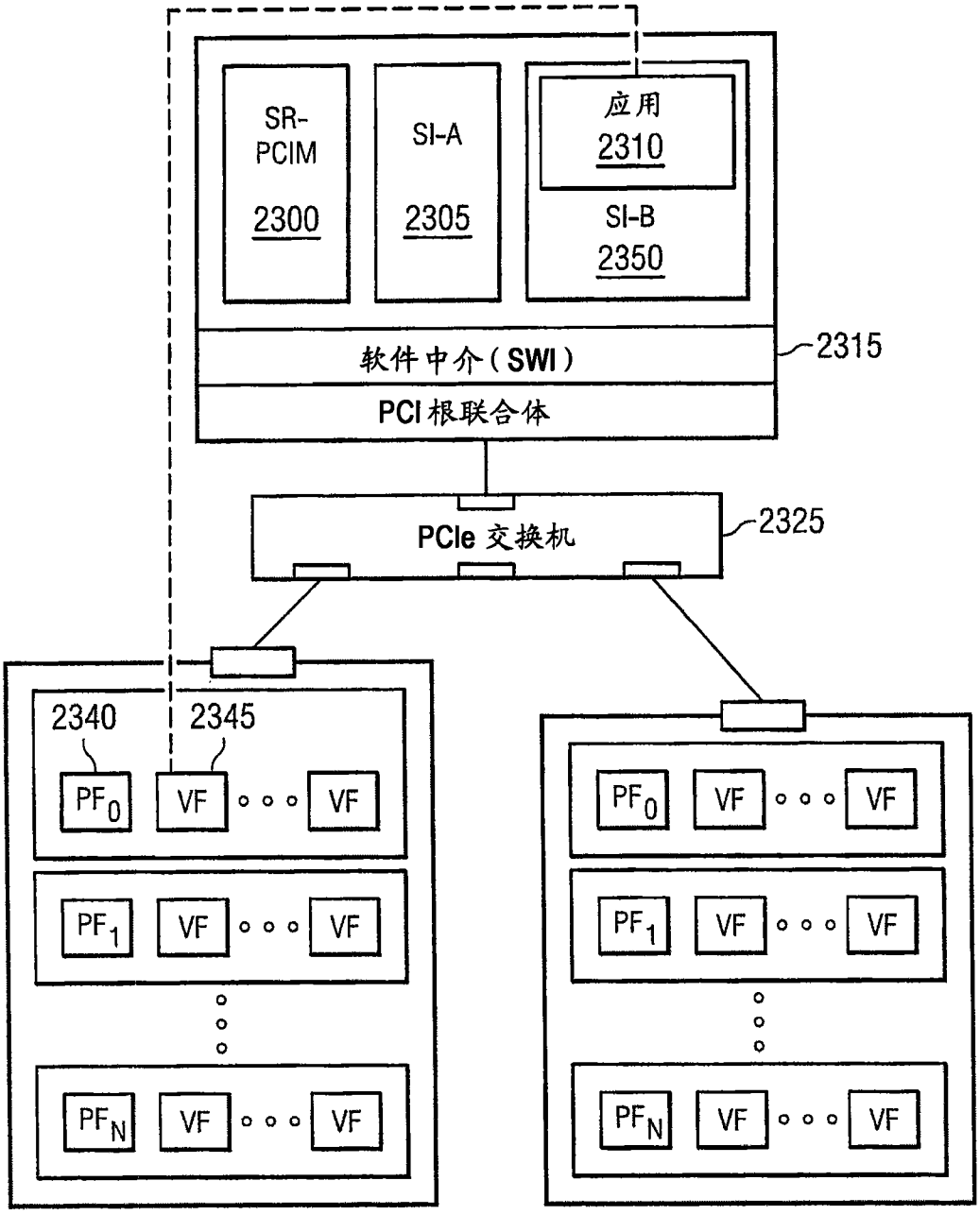

SR-PCIM
2300
SI-A
2305
应用
2310
SI-B
2350
软件中介(SWI)
2315
PCI根联合体
PCIe交换机
2325
2340
2345
PF0
VF
PF1
PFN

图 23B

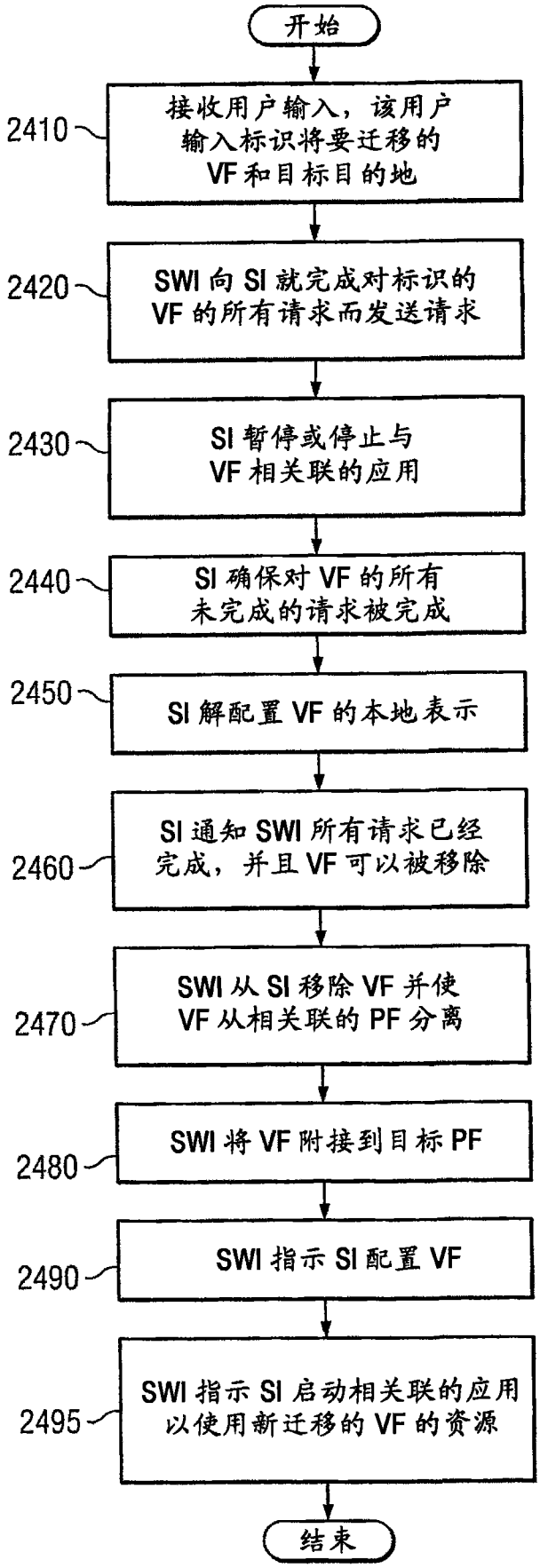

开始
2410
接收用户输入，该用户输入标识将要迁移的VF和目标目的地
2420
SWI向SI就完成对标识的VF的所有请求而发送请求
2430
SI暂停或停止与VF相关联的应用
2440
SI确保对VF的所有未完成的请求被完成
2450
SI解配置VF的本地表示
2460
SI通知SWI所有请求已经完成，并且VF可以被移除
2470
SWI从SI移除VF并使VF从相关联的PF分离
2480
SWI将VF附接到目标PF
2490
SWI指示SI配置VF
2495
SWI指示SI启动相关联的应用以使用新迁移的VF的资源
结束

图 24

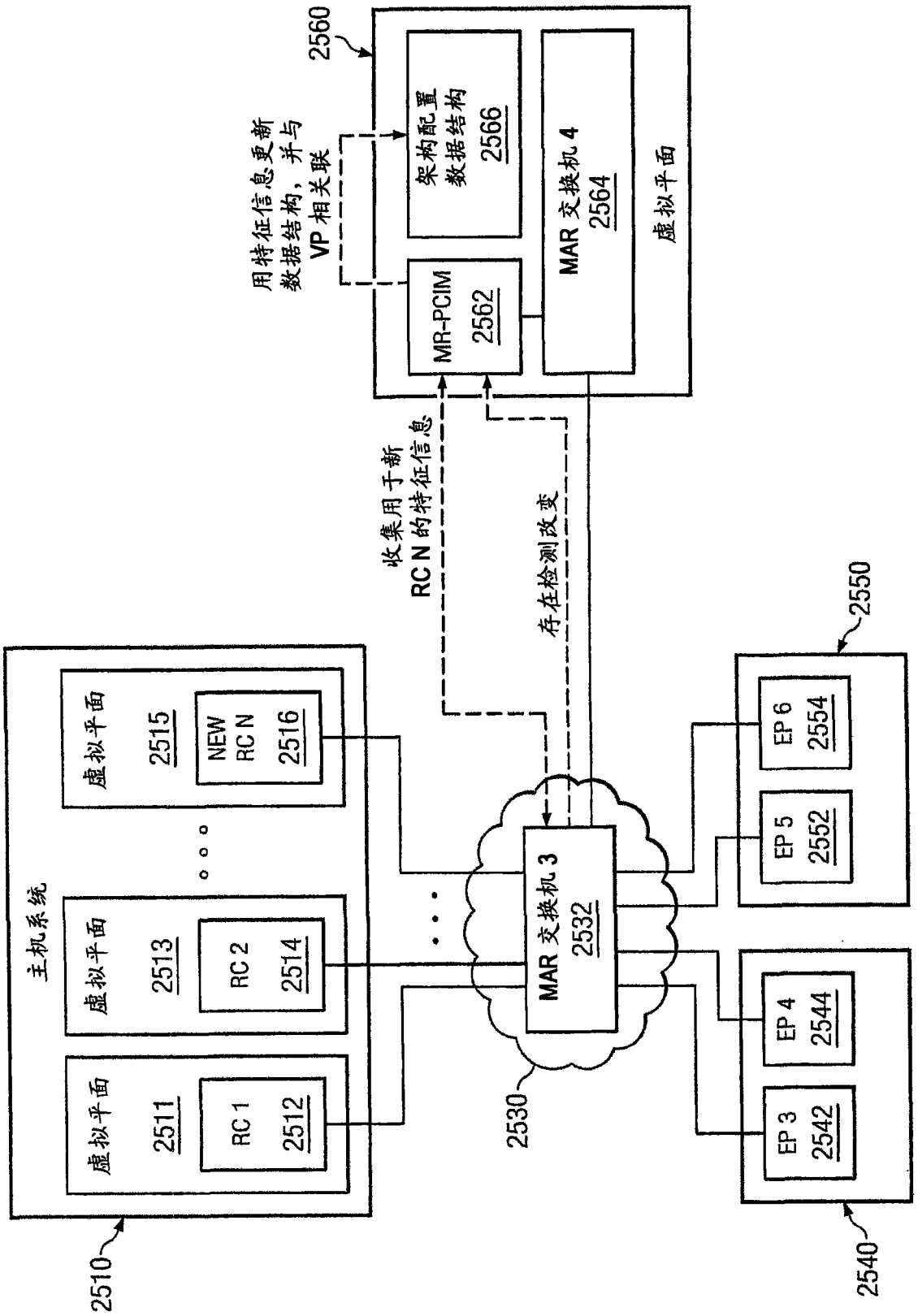

2560
架构配置
数据结构
2566
MAR 交换机 4
2564
虚拟平面
MR-PCIM
2562
用特征信息更新
数据结构，并与
VP 相关联
收集用于新
RC N 的特征信息
存在检测改变
2550
EP 6
2554
EP 5
2552
EP 4
2544
EP 3
2542
2540
MAR 交换机 3
2532
2530
主机系统
虚拟平面
2515
NEW
RC N
2516
虚拟平面
2513
RC 2
2514
虚拟平面
2511
RC 1
2512
2510

图 25

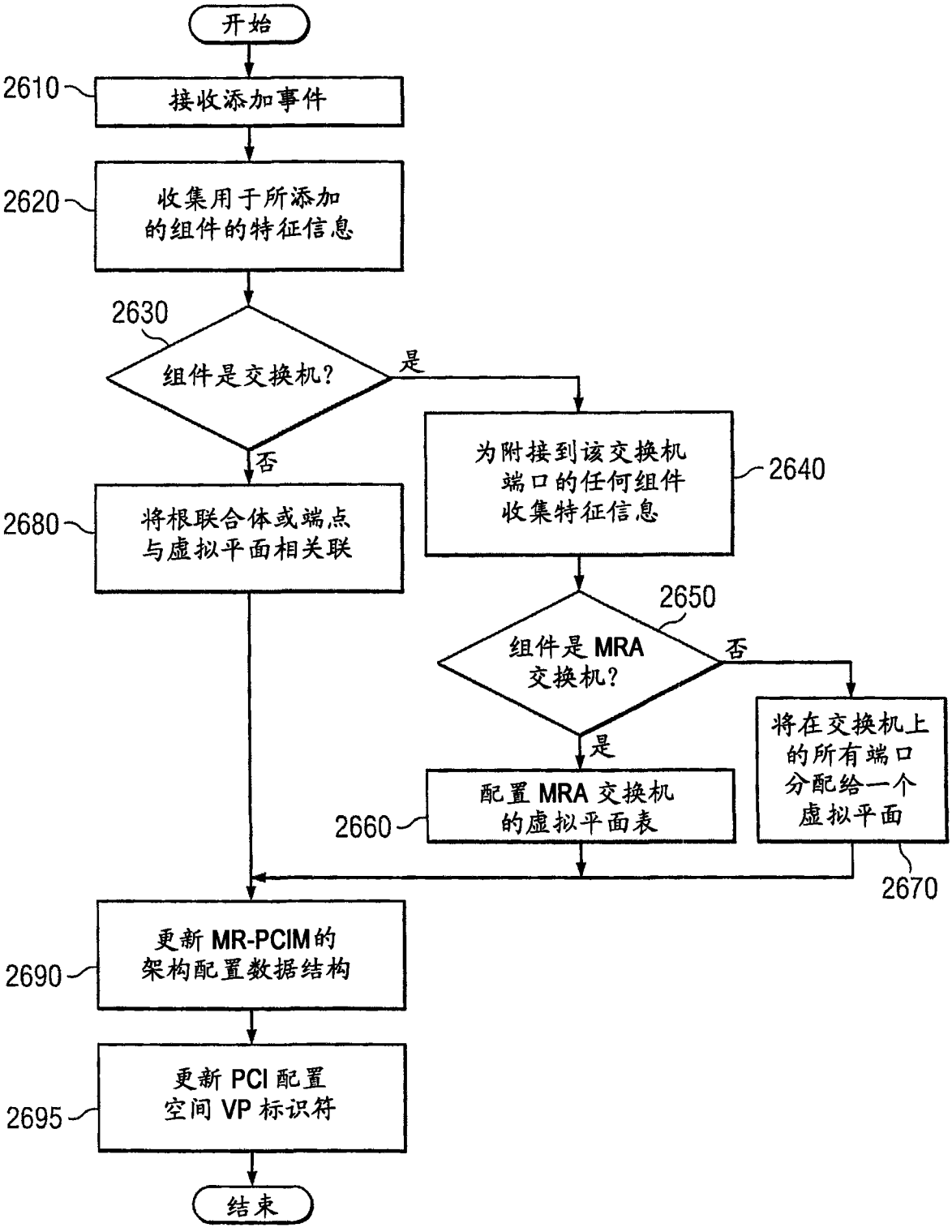

开始
2610
接收添加事件
2620
收集用于所添加的组件的特征信息
2630
组件是交换机?
是
否
2640
为附接到该交换机端口的任何组件收集特征信息
2650
组件是 MRA 交换机?
否
是
2660
配置 MRA 交换机的虚拟平面表
将在交换机上的所有端口分配给一个虚拟平面
2670
2680
将根联合体或端点与虚拟平面相关联
2690
更新 MR-PCIM 的架构配置数据结构
2695
更新 PCI 配置空间 VP 标识符
结束

图 26

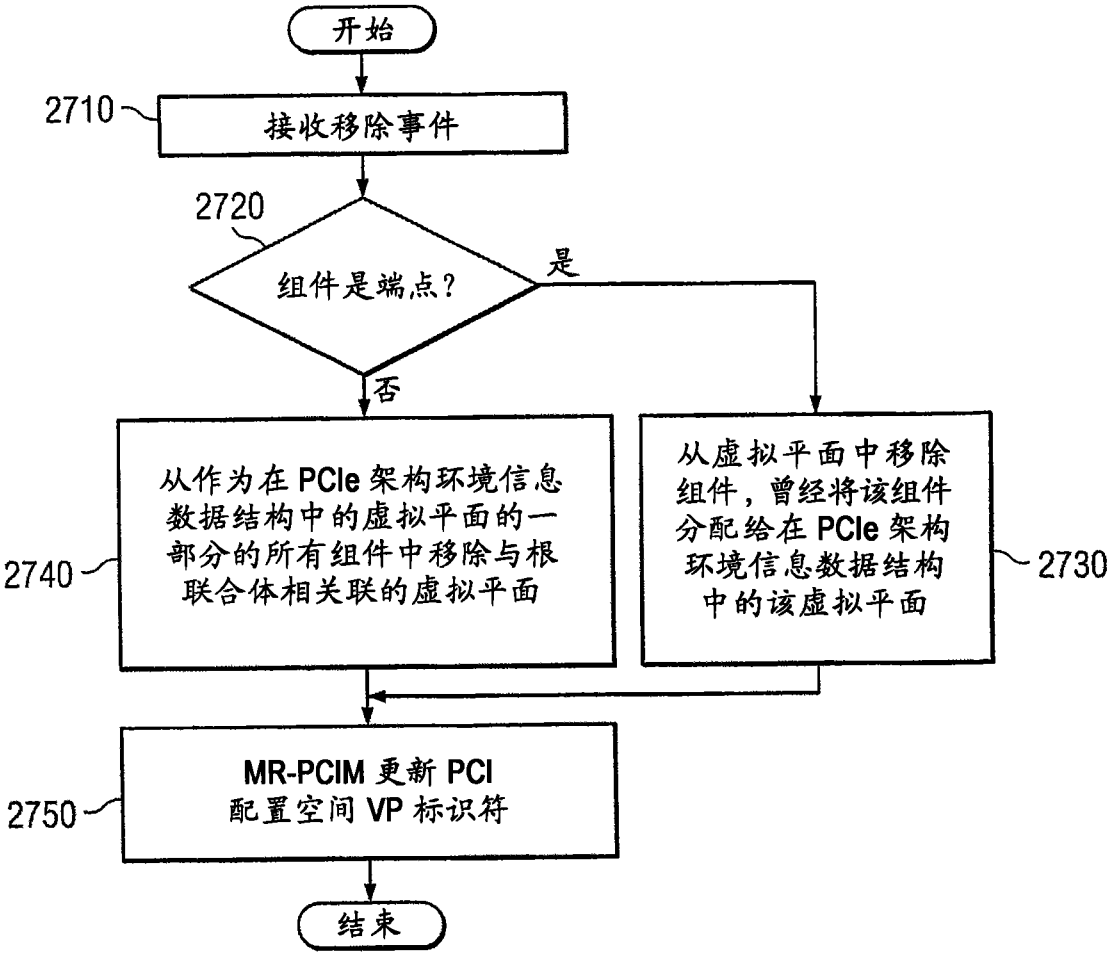

开始
2710
接收移除事件
2720
组件是端点?
是
否
从作为在 PCIe 架构环境信息数据结构中的虚拟平面的一部分的所有组件中移除与根联合体相关联的虚拟平面
2740
从虚拟平面中移除组件，曾经将该组件分配给在 PCIe 架构环境信息数据结构中的该虚拟平面
2730
MR-PCIM 更新 PCI 配置空间 VP 标识符
2750
结束

图 27