使用 PCIe 交换网结构在多主机系统中优化资源部署

Microchip Technology Inc.
固件工程技术顾问
Vincent Haché

越来越多的数据中心和其他高性能计算环境开始使用 GPU，因为 GPU 能够快速处理深度学习和机器学习应用中生成的大量数据。不过，就像许多可提高应用性能的新型数据中心创新一样，这项创新也暴露出新的系统瓶颈。在这些应用中，用于提高系统性能的新兴架构涉及通过一个 PCIe®结构在多个主机之间共享系统资源。

PCIe 标准（特别是其基于树的传统层级）会限制资源共享的实现方式（和实现程度）。不过，可以实现一种低延时的高速结构方法，这种方法允许在多个主机之间共享大量 GPU 和 NVMe SSD，同时仍支持标准系统驱动程序。

PCIe 结构方法采用动态分区和多主机单根 I/O 虚拟化（SR-IOV）共享。各 PCIe 结构之间可直接路由点对点传输。这样便可为点对点传输提供最佳路由，减少根端口拥塞，并且更有效地平衡 CPU 资源的负载。

传统上，GPU 传输必须访问 CPU 的系统存储器，这会导致端点之间发生存储器共享争用。 当 GPU 使用其共享的存储器映射资源而不是 CPU 存储器时，它可以在本地提取数据，无需先通过 CPU 传递数据。这消除了跳线和链路以及由此产生的延时，从而使 GPU 能够更高效地处理数据。

PCIe 的固有限制

PCIe 主层级是一个树形结构，其中的每个域都有一个根联合体，从该点可扩展到“叶子”，这些“叶子”通过交换网和桥接器到达端点。链路的严格层级和方向性给多主机、多交换网系统带来了成本高昂的设计要求。

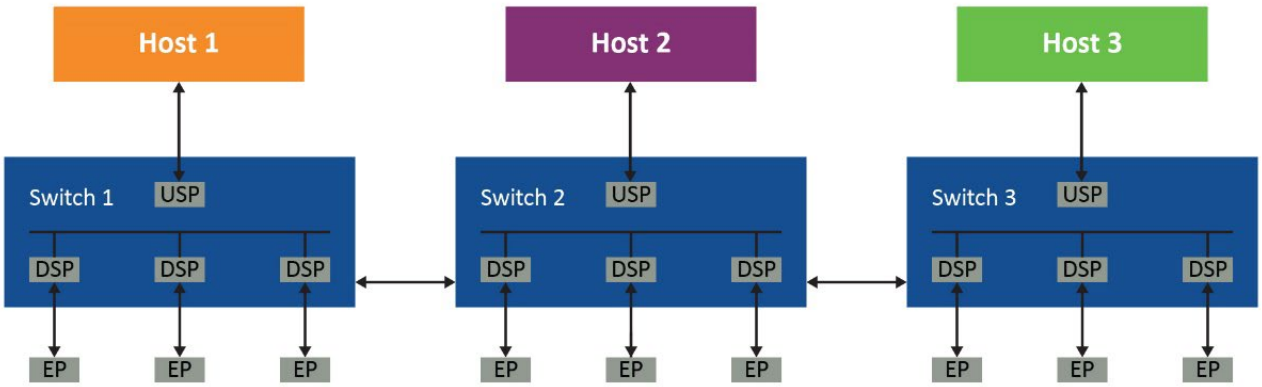

图 1——多主机拓扑

以图 1 所示的系统为例。要符合 PCIe 的层级，主机 1 必须在交换网 1 中有一个专用的下行端口，该端口连接到交换网 2 中的专用上行端口。它还需要在交换网 2 中有一个专用的下行端口，该端口连接到交换网 3 中的专用上行端口，依此类推。主机 2 和主机 3 也有类似的要求，如图 2 所示。

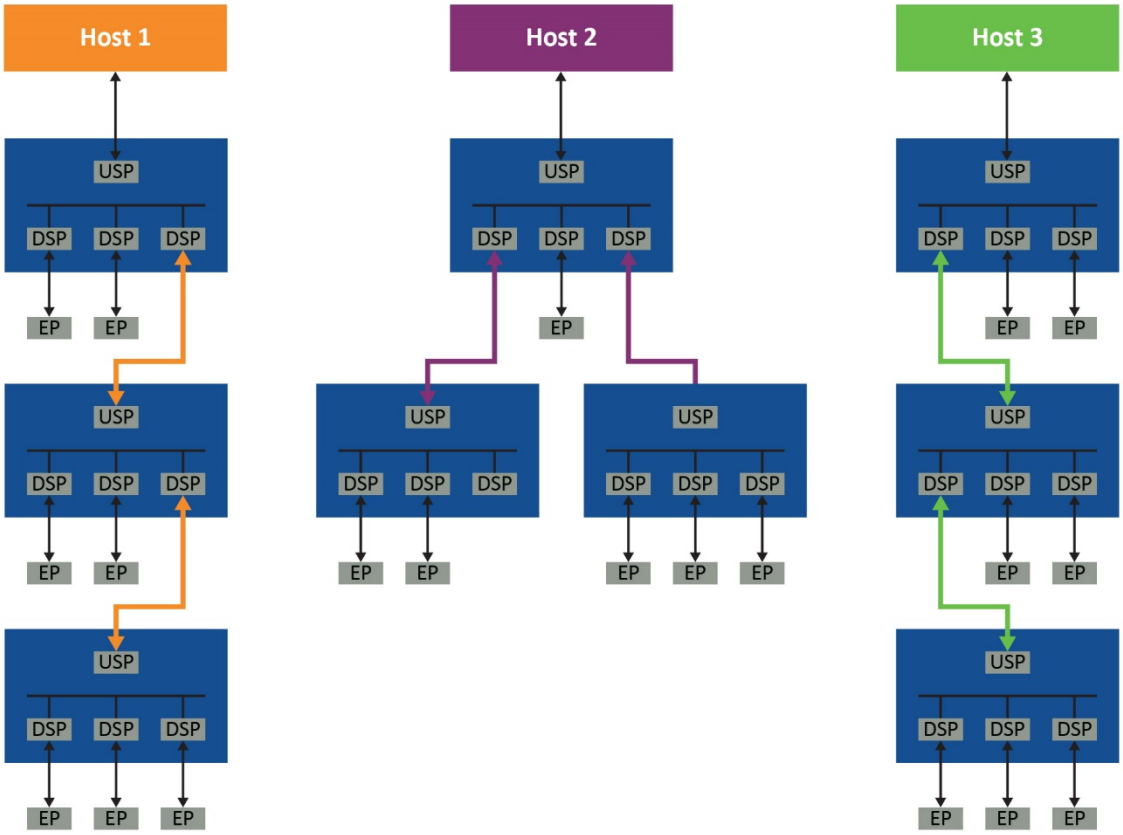

图 2——每个主机的层级要求

即使是基于 PCIe 树形结构的最基本系统，也需要各交换网之间有三个链路专用于每个主机的 PCIe 拓扑。而且，由于主机之间无法共享这些链路，因此系统会很快变得极为低效。

此外，符合 PCIe 的典型层级只有一个根端口，而且尽管“多根 I/O 虚拟化和共享”规范中支持多个根，但它会使设计更复杂，并且当前不受主流 CPU 支持。结果会造成未使用的 PCIe 设备（即端点）滞留在其分配到的主机中。不难想象，这在采用多个 GPU、存储设备及其控制器以及交换网的大型系统中会变得多么低效。

例如，如果第一个主机（主机 1）已经消耗了所有计算资源，而主机 2 和 3 未充分利用资源，则显然希望主机 1 访问这些资源。但主机 1 无法这样做，因为这些资源在它的层级域之外，因此会发生滞留。非透明桥接（NTB）是这种问题的一个潜在解决方案，但由于每种类型的共享 PCIe 设备都需要非标准驱动程序和软件，因此这同样会使系统变得复杂。更好的方法是使用 PCIe 结构，这种结构允许标准 PCIe 拓扑容纳多个可访问每个端点的主机。

实施方法

系统使用一个 PCIe 结构交换网（本例中为 Microchip Switchtec® PAX 系列的成员）在两个独立但可透明互操作的域中实现：即包含所有端点和结构链路的结构域以及每个主机专用的主机域（图 3）。主机通过在嵌入式 CPU 上运行的 PAX 交换网固件保留在单独的虚拟域中，因此，交换网将始终显示为具有直连端点的标准单层 PCIe 设备，而与这些端点出现在结构中的位置无关。

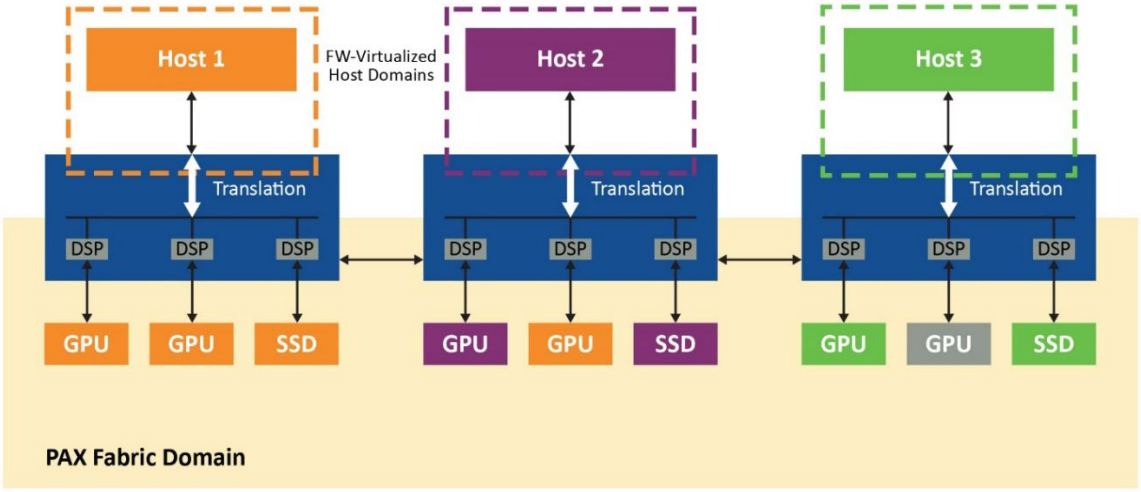

图 3——每个结构的独立域

来自主机域的事务会在结构域中转换为 ID 和地址，反之，结构域中通信的非分层路由也是如此。这样，系统中的所有主机便可共享连接交换网和端点的结构链路。交换网固件会拦截来自主机的所有配置平面通信（包括 PCIe 枚举过程），并使用数量可配置的下行端口虚拟化一个符合 PCIe 规范的简单交换网。

当所有控制平面通信都路由到交换网固件进行处理时，数据平面通信直接路由到端点。其他主机域中未使用的 GPU 不再滞留，因为它们可以根据每个主机的需求动态分配。结构内支持点对点通信，这使其能够适应机器学习应用。当以符合 PCIe 规范的方式向每个主机提供功能时，可以使用标准驱动程序。

操作方法

为了解这种方法的工作原理，我们以图 4 中的系统为例，该系统由两个主机（主机 1 采用 Windows®系统，主机 2 采用 Linux®系统）、四个 PAX PCIe 结构交换网、四个 Nvidia M40 GPGPU 和一个支持 SR-IOV 的 Samsung NVMe SSD 组成。在本实验中，主机运行代表实际机器学习工作负载的通信，包括 Nvidia 的 CUDA 点对点通信基准测试实用程序和训练 cifar10 图像分类的 TensorFlow 模型。嵌入式交换网固件处理交换网的低级配置和管理，系统由 Microchip 的 ChipLink 调试和诊断实用程序管理。

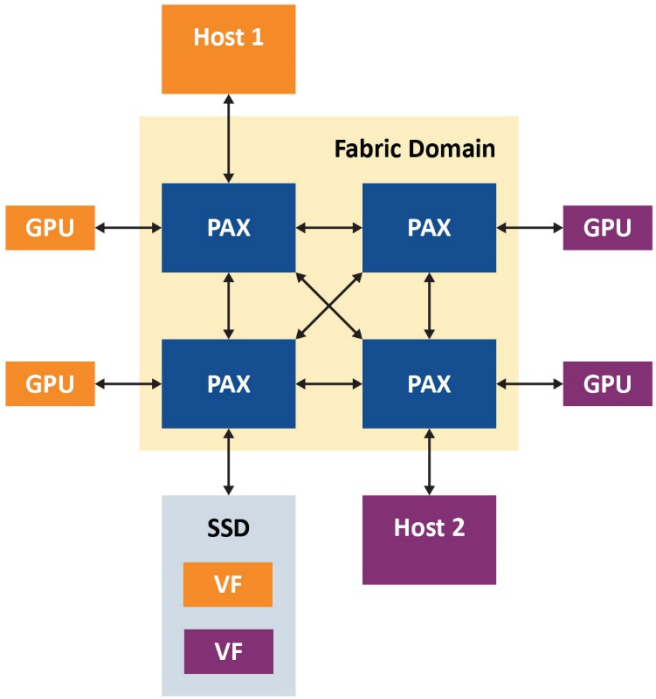

图 4：双主机 PCIe 结构引擎

四个 GPU 最初分配给主机 1，PAX 结构管理器显示在结构中发现的所有设备，其中 GPU 绑定到 Windows 主机。但是，主机上的结构不再复杂，所有 GPU 就像直接连接到虚拟交换网一样。随后，结构管理器将绑定所有设备，Windows 设备管理器将显示 GPU。主机将交换网视为下行端口数量可配置的简单物理 PCIe 交换网。

一旦 CUDA 发现了四个 GPU，点对点带宽测试就会显示单向传输速率为 12.8 GBps，双向传输速率为 24.9 GBps。这些传输直接跨过 PCIe 结构，而无需通过主机。如果运行用于训练 Cifar10 图像分类算法的 TensorFlow 模型并使工作负载分布在全部四个 GPU 上，则可以将两个 GPU 释放回结构池中，将它们与主机解除绑定。这样可以释放其余两个 GPU 来执行其他工作负载。与 Windows 主机一样，Linux 主机也将交换网视为简单的 PCIe 交换网，无需自定义驱动程序，而 CUDA 也可以发现 GPU，并在 Linux 主机上运行 P2P 传输。性能类似于使用 Windows 主机实现的性能，如表 1 所示。

表 1：GPU 点对点传输带宽

事务类型	主机 1 平均带宽	主机 2 平均带宽
单向 P2P	12.8 GBps	12.7 GBps
双向 P2P	24.9 GBps	24.6 GBps

下一步是将 SR-IOV 虚拟功能连接到 Windows 主机，PAX 将此类功能以标准物理 NVM 设备的形式提供，以便主机可以使用标准 NVMe 驱动程序。此后，虚拟功能将与 Linux 主机结合，并且新的 NVMe 设备将出现在模块设备列表中。本实验的结果是，两个主机现在都可以独立使用其虚拟功能。

务必注意的是，虚拟 PCIe 交换网和所有动态分配操作都以完全符合 PCIe 规范的方式呈现给主机，以便主机能够使用标准驱动程序。嵌入式交换网固件提供了一个简单的管理接口，这样便可通过成本低廉的外部处理器来配置和管理 PCIe 结构。设备点对点事务默认情况下处于使能状态，不需要外部结构管理器进行额外配置或管理。

总结

PCIe 交换网结构是一种能够充分利用 CPU 巨大性能的绝佳方法，但 PCIe 标准本身存在一些障碍。不过，可以通过使用动态分区和多主机单根 I/O 虚拟化共享技术来解决这些难题，以便可以将 GPU 和 NVMe 资源实时动态分配给多主机系统中的任何主机，从而满足机器学习工作负载不断变化的需求。