UALink 200G 1.0

Scale-Up 互联技术白皮书

作者: Nathan Kalyanasundharam
AMD Corporate Fellow &
UALink 联盟技术工作组联合主席

ULTRA
ACCELERATOR
LINK™

Notice:

This technical whitepaper was originally prepared in English. A Chinese version of the white paper was authorized by UALink Consortium and translated by Lu Lu and Dajiang Zhang from AMD Product (China) Co. Ltd.

Any translation of this document is provided for only reference and convenience. In the event of any conflict, discrepancy, omission, or ambiguity between the English version and the translated version, the English version shall take precedence and control for all purposes of interpretation and application.

本中文版本经 UALink 联盟授权，由 AMD 产品（中国）有限公司的卢璐与张大江翻译。

本技术规范原文以英文撰写。本文件的任何中文或其他语言译本仅为参考和便利之用。在实际运用或理解中，若英文版本与任何译本之间存在冲突、差别、遗漏或歧义，均以英文版本为准。

ULTRA ACCELERATOR LINK™

目录

1 引言

Ultra Accelerator Link™ (UALink™) 作为全新推出的行业标准，致力于为下一代人工智能工作负载提供纵向扩展 (scale-up) 互连解决方案。UALink 200G 1.0 规范最初由 UALink 联盟发起成员（包括阿里巴巴、AMD、苹果、Astera Labs、亚马逊云科技、思科、谷歌、HPE、英特尔、Meta、微软和新思科技）联合开发，目前已获得 70 多家贡献者和采用者的支持与使用。

随着 AI 模型的飞速迭代，其对计算能力、内存容量及互连性能的需求正持续攀升。在包含数百个加速器的 POD 中分发人工智能模型时，纵向扩展（Scale-up）解决方案至关重要。提供可靠的 Scale-up 解决方案所需的成本与复杂性，已成为整个行业的一大负担。目前，行业对建立基于标准的 Scale-up 网络解决方案以应对训练和推理工作负载的需求不断增长。UALink 联盟的使命是制定一项开放标准，以提供可扩展、高性能、弹性且经济高效的 Scale-up 连接网络解决方案。

UALink 旨在减少互联解决方案对芯片面积的占用、加速器互访延迟及整体功耗，同时兼顾互联带宽的高效使用。其核心特点包括：

- 提升双向内存访问的链路效率，以实现最大化数据带宽。
- 借助现有的以太网基础设施（涵盖线缆、连接器、重定时器及管理软件的使用），降低总体拥有成本（TCO）。
- 采用内存语义，支持直接读写（load, store）、原子事务，并与主机连接的内存、本地加速器内存和远程加速器内存保持相同的排序模型，从而降低软件复杂性。

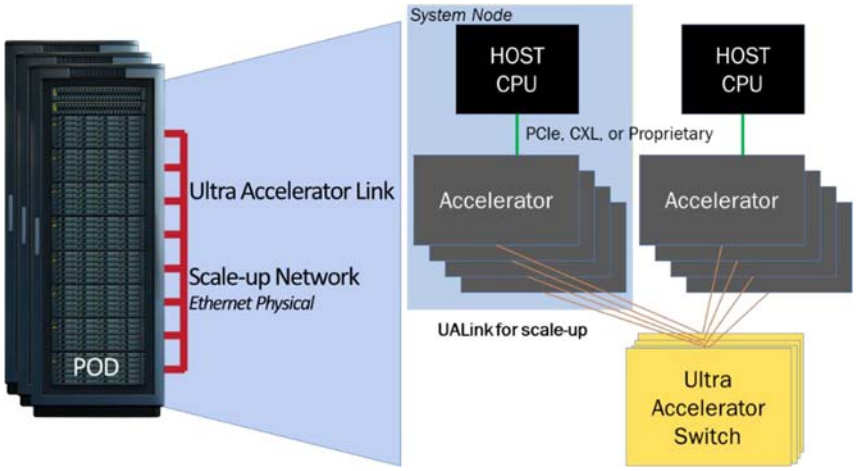

图 1：纵向扩展(Scale-up)机架基础设施中的多节点和多平面交换

UALink 1.0 规范详细定义了支持加速器间低延迟通信的协议与接口，为GPU 等加速卡服务器集群提供了高效可扩展、高带宽、低延迟的互连性能，精准匹配人工智能发展过程中计算能力、内存带宽及容量的增长需求。

2 概述

UALink 1.0 规范支持每通道最高 200 GT/s 的数据传输速率。UALink的物理层是以太网物理层设计。考虑到以太网物理层进行前向纠错码 (FEC) 和编码所带来的带宽损耗，其目标信号传输速率为 212.5 GT/s。UALink 物理通道可支持多种宽度的配置和组合：最高 4x 单通道链路 (x1 Link) 、或者 2x 双通道链路 (x2 Link) 或 1x 四通道链路 (x4 Link) 。每四条物理通道组合在一起构成一个UALink的基本单元组，在发送 (TX) 和接收 (RX) 方向上各提供最大 800 Gbps 的带宽。系统中加速器的数量和分配给每个加速器的带宽可以实现自由配置和扩展，以满足各种 AI 应用的需求。图 2 展示了 UALink 的主要特性和目标。

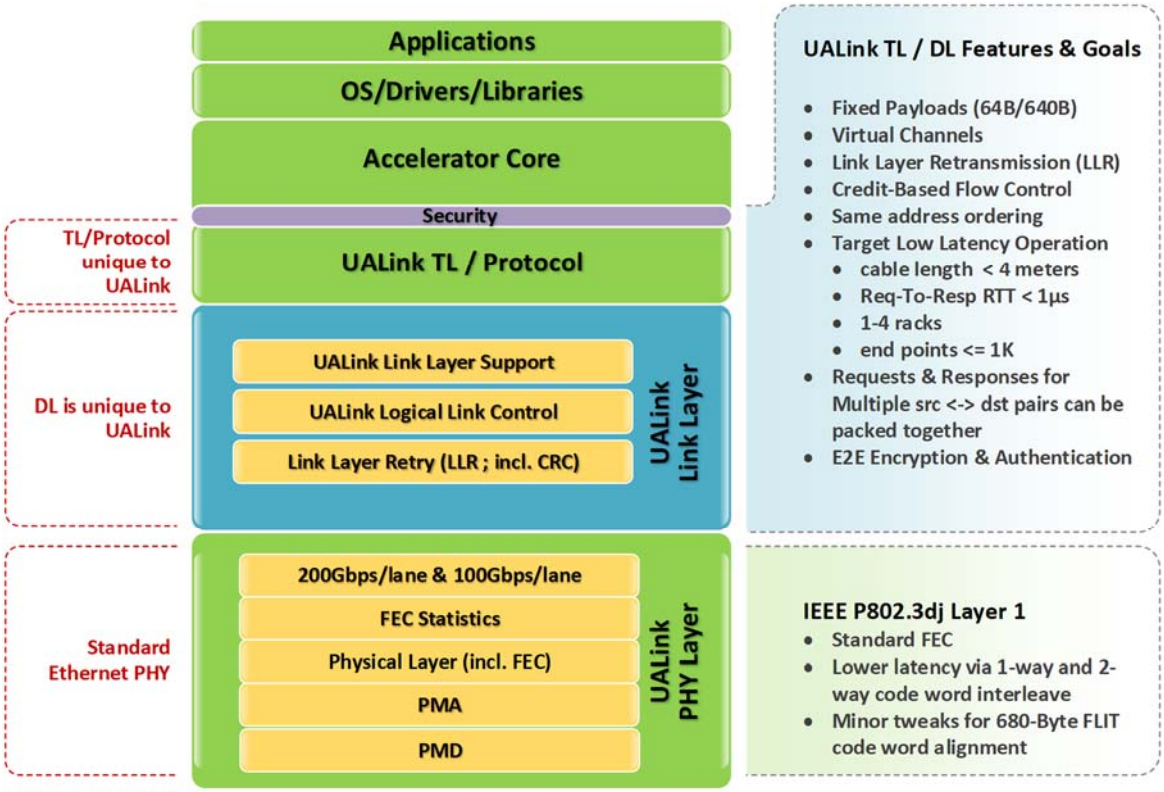

图 2： UALink 特性与目标

图3是一个多节点系统的示例，每个服务器节点均配备 1 个主机处理器和 4 个加速器。整个系统总共包含 'M' 个加速器，每个加速器有 'N' 个端口。加速器的每个端口连接到不同的UALink Switch（ULS），ULS的每个端口连接到一个不同的加速器，以此可实现流量的均匀分配。在UALink 1.0 一层交换机的架构下，加速器的端口数N与所连接的交换机总数相匹配，而交换机端口数量M与其连接的加速器总数量相匹配。通过ULS相互连接的加速器集群共同构成一个Scale-up AI POD (Point of Delivery) 。UALink 1.0 可支持多达1024个加速器端点的扩展，每个加速器会被分配一个唯一的 10 位路由标识符。

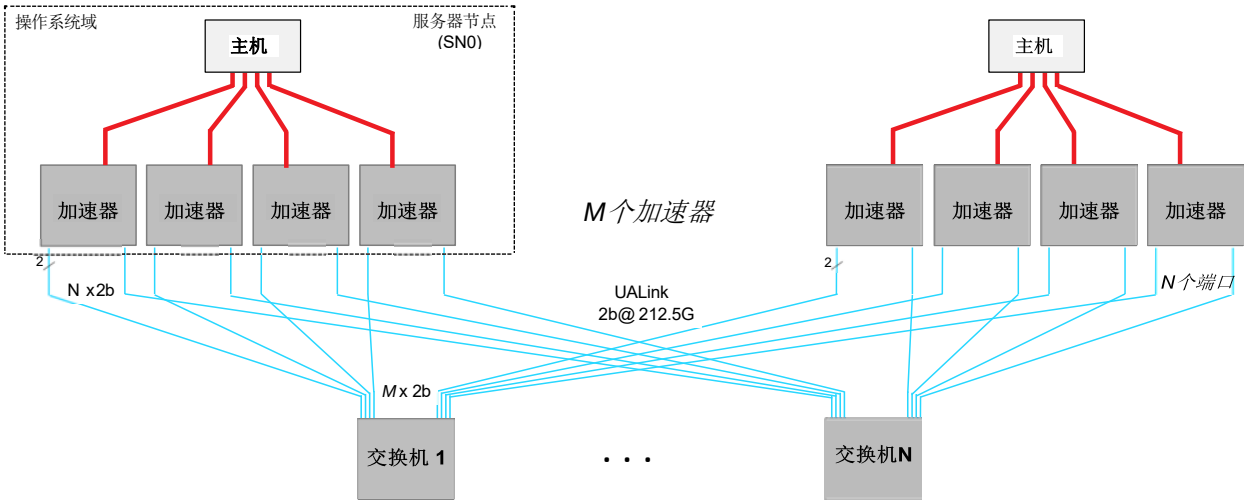

图 3：基于 UALink 的多加速器系统

在实际应用中，一个POD可以被划分为多个虚拟 POD (Virtual POD)。Virtual POD 是 由POD中的一个或多个加速器组成的分组，组内的加速器可相互通信，但无法与组外的其他加速器进行通信。ULS 交换机需要支持相应的端口分区的配置机制，通过将每个交换机上的端口划分为不重叠的子集，可实现 POD 到虚拟 POD 的划分。每个子集内的端口能相互通信，而无法与子集外的端口建立连接。值得注意的是，无论虚拟 POD 如何划分，POD 内的所有加速器都拥有唯一的加速器 ID。

3 UALink 分层架构

如图4所示，一个完整的 UALink 分层架构包括：

- 协议层（Protocol Layer）- UPLI
- 事务层（Transaction Layer）- TL
- 数据链路层（Data Link Layer）- DL
- 物理层（Physical Layer）- PL

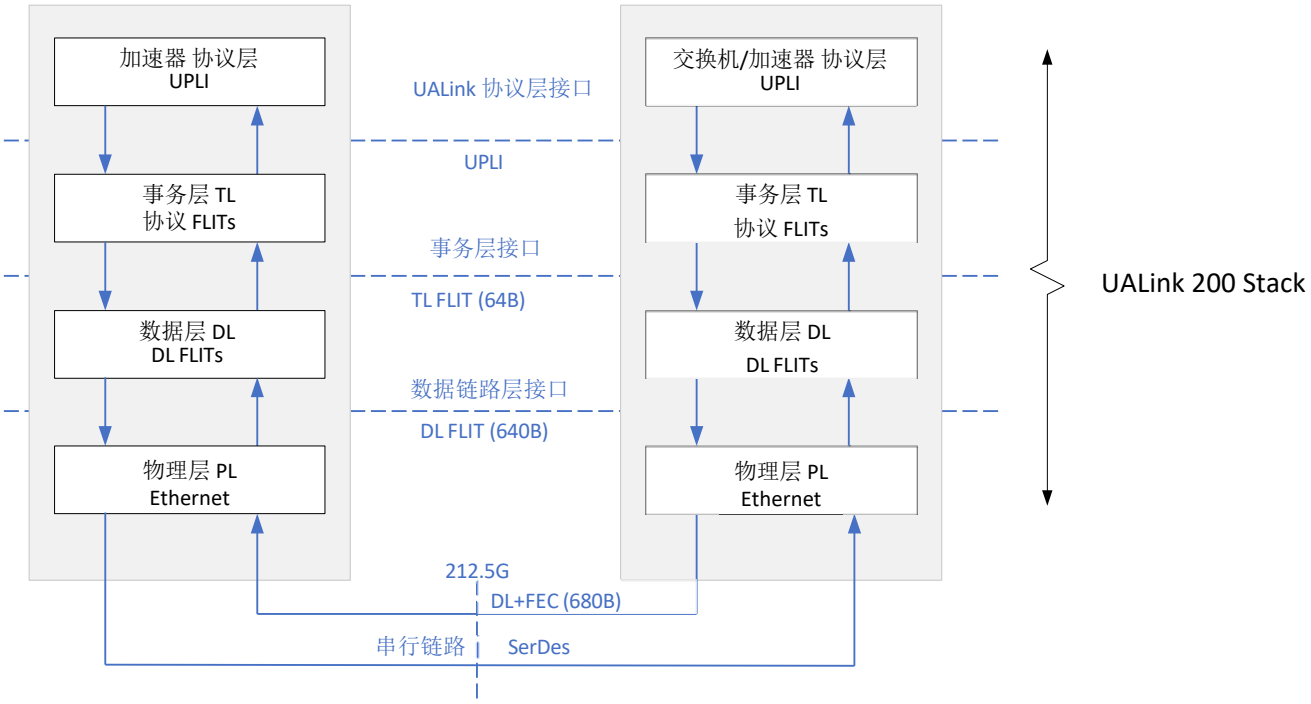

图4： UALink 分层架构图

在AI POD执行训练或者推理任务时，计算单元需要的数据可能分布在不同的加速器的内存中。在使用UALink技术互联的 Scale-up POD 中，加速器之间可以通过UALink直接进行基于内存语义的数据传输。UALink的协议层是对称结构，每个UALink基本单元组的发送和接收路径由相同的通道构成。加速器之间的数据收发需要穿过 UALink 的各个功能层。

每个UALink的协议层包含一个独立的UPLI (UALink Protocol Layer Interface) 发起方 (Originator) 和一个独立的 UPLI 完成方 (Completer)，各自连接到传输层(TL)。传输层将UPLI 协议下各通道的传输内容封装为 传输层 Flit，并传递给数据链路层 (DL)做链路层封装。同样，传输层（TL） 也会从 数据链路层（DL） 接收 TL Flit 并将其解包为 UPLI 协议。发起方UPLI接口发起针对远程加速器的请求并接收响应，完成方UPLI接口接收来自对端加速器的请求并返回响应。

UALink 数据链路层（DL）接收多个传输层 Flit，为其添加循环冗余校验（CRC）保护和Header信息以形成数据链路层Flit，并将该数据链路层 Flit 传递至 UALink 物理层（PL）。同样，数据链路层也会从物理层接收数据链路层 Flit，剥离其中的CRC和Header信息，形成传输层 Flit 并传递给传输层（TL）。UALink 物理层（PL）接收数据链路层 Flit 后，生成经前向纠错（FEC）编码的码字，这些码字经串行化处理后通过串行链路传输。在接收方向，物理层还能从交换机或加速器上的串行链路接收带 FEC 的串行化码字，执行 FEC 解码并将其转换为数据链路层（DL） Flit。

3.1 物理层 (PL)

基于 IEEE802.3 以太网物理层，UALink 可以支持 212.5G 串行速率（200GBASE-KR1/CR1, 400GBASE-KR2/CR2, 800GBASE-KR4/CR4）或者较低速的106.25G 速率（100GBASE-KR1/CR1, 200GBASE-KR2/CR2, 400GBASE-KR4/CR4）下的一条、二条或四条通道。

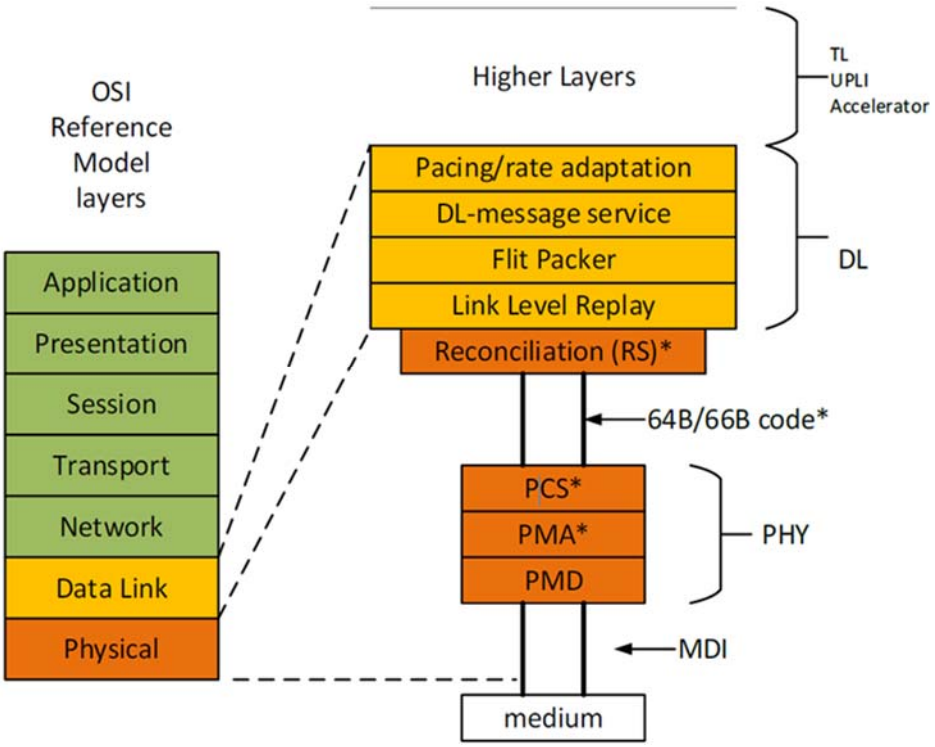

图5：UALink 物理层

图 5 中的 * 表示此处对 IEEE 802.3 有修改。UALink 架构的 PCS（物理编码子层）/PMA（物理介质适配接口）层新增了减少交织的codeword交织模式，以降低突发错误纠正能力为代价来实现更小的 FEC 延迟。相对于 IEEE 802.3, UALink物理层的PMD（物理介质相关子层），Auto Negotiation(自动协商) 和 Link Training (链路训练) 功能未作出任何修改。UALink 使用了 IEEE 802.3 的 64B/66B 编码。

UALink PCS 和 RS（协调层 - PCS 和 DL 之间的接口）将640字节（包含CRC）的DL Flit 精准适配到 Reed Solomon (544, 514) 码字中，实现了延迟和 Replay Flit 的最小化。

3.2 数据链路层 (DL)

数据链路层位于事务层和物理层之间。数据链路层（DL）将来自事务层的 64 字节 Flit 打包成 640 字节的 Flit 传递给物理层。此外，数据链路层还在链路对端间提供消息服务。此消息服务可用于通知对端事务层的速率、查询链路对端设备及端口 ID 等功能。同时，数据链路层还能在链路对端间提供类似于通用异步收发传输的消息通信方式，服务于固件（Firmware）通信。链路级重传以 640 字节 Flit 为单位进行，32 位循环冗余校验（CRC）会经过计算和校验，并作为 640 字节 Flit 的一部分被包含在内。

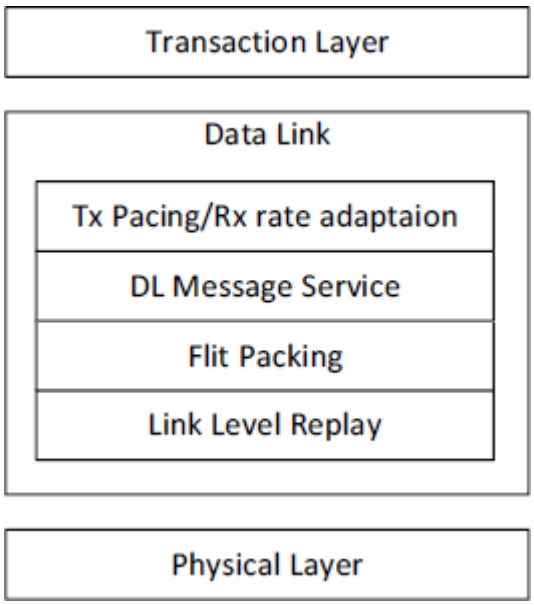

图 6：数据链路层框图

3.3 事务层(TL)

Scale-up 互联网络需要支持内存语义，以GPU为例的加速器的内存访问指令一般是以Cacheline Size为基本操作单元。同时，加速器需要在协议层提供目标加速器ID来方便交换机进行路由。UALink的协议层基于这些需求提供了高效便捷的UPLI (UALink Protocol Level Interface)接口。UPLI 接口是一种多通道和基于Credit流控的接口协议，包含请求，数据，写回复和读回复及相应读数据通道。同时，UPLI 协议具备内置灵活性，允许供应商为同类型加速器之间的通信创建自定义协议消息，且无需对 UALink 交换机做任何修改。

UALink的事务层（TL）负责将两个UPLI接口（Originator / Completer）接收方向通道的协议传输转换为以64字节为单位的传输层 Flit。此外，事务层还会将从数据链路层（DL）接收到的TL Flit，重新转换为 UPLI 接口上的 UPLI 通道事务。

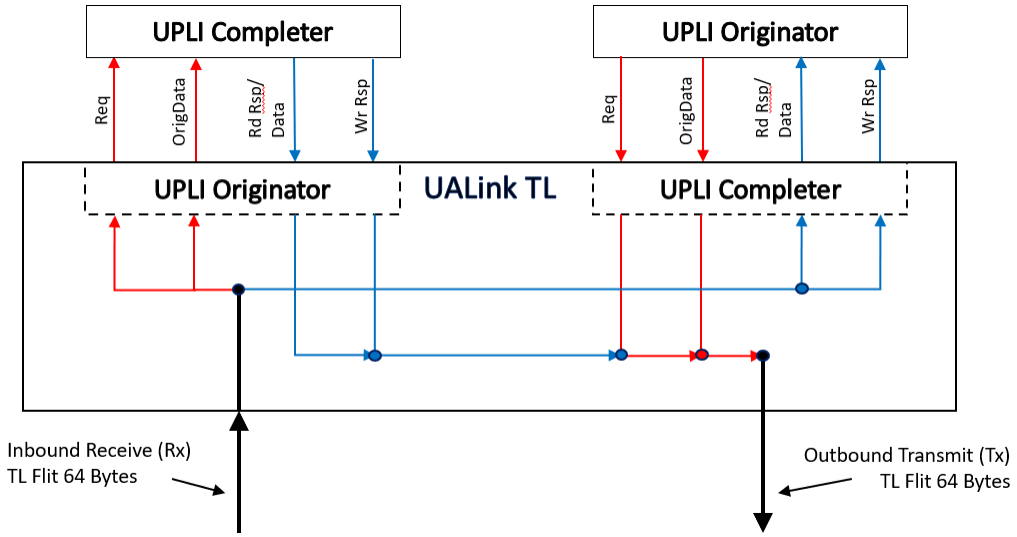

图7：UALink 事务层接口

图 7 示意性地展示了事务层（TL）接口及其在两个 UPLI 接口的各通道之间的连接关系，同时也呈现了这些接口与发送（Tx）和接收（Rx）传输层 Flit 的关联。由于TL接口具有对称性，接收Flit和发送Flit的格式完全相同。每个 64 字节的传输层 Flit 包含同方向 多个UPLI 通道的信息或者数据。UALink的事务层还创新性地支持流式地址缓存，用以压缩地址域。这使得 TL 能够实现非常高的双向协议效率。

图 8 展示了一个具体的 TL Flit 打包示例，其中包括五个写请求、五个写响应和一个流控制包。整体需要 21 个 TL Flit 的传输，协议传输效率可以达到 95.2% (20/21) 。压缩后的每个写请求为 8 字节，压缩后的写响应和流控制各为 4 字节。

	64-byte TL Flit															
	Upper TL Half-Flit								Lower TL Half-Flit							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Flit																
0	Req0.Data.0								CWReq2		CWReq1		CWR B	CWR A	CWReq0	
1	Req0.Data.2								Req0.Data.1							
2	Req0.Data.4								Req0.Data.3							
3	Req0.Data.6								Req0.Data.5							
4	Req1.Data.0								Req0.Data.7							
5	Req1.Data.2								Req1.Data.1							
6	Req1.Data.4								Req1.Data.3							
7	Req1.Data.6								Req1.Data.5							
8	Req2.Data.0								Req1.Data.7							
9	Req2.Data.2								Req2.Data.1							
10	Req2.Data.4								Req2.Data.3							
11	Req2.Data.6								Req2.Data.5							
12	Req2.Data.7								CWReq4		CWReq3		CWR E	CWR D	CWR C	FC
13	Req3.Data.1								Req3.Data.0							
14	Req3.Data.3								Req3.Data.2							
15	Req3.Data.5								Req3.Data.4							
16	Req3.Data.7								Req3.Data.6							
17	Req4.Data.1								Req4.Data.0							
18	Req4.Data.3								Req4.Data.2							
19	Req4.Data.5								Req4.Data.4							
20	Req4.Data.7								Req4.Data6							

图 8：写传输的最高效率示例（256字节传输请求，伴随写请求压缩）

4 安全特性

UALink 的安全特性称为 UALinkSec，旨在保护 UALink 网络和交换机上的流量免受物理攻击者的威胁 —— 攻击者可能在攻击时直接在场，也可能通过植入设备（例如中介器）以窥探或篡改 UALink 流量。此外，在支持机密计算（CC）的平台中，UALinkSec 可保护 UALink 网络和交换机上的租户数据，使其免受基础设施提供商以及同一UALink POD 中其他共存租户的影响。机密计算意味着平台上存在可信执行环境（TEE）（例如英特尔 TDX、AMD SEV 和 ARM CCA）。由租户控制的 TEE 负责 UALinkSec 的配置。启用后，UALinkSec 可提供数据机密性以及可选的数据完整性（包括重放保护）。

UALinkSec 支持对所有 UPLI 协议通道（请求、读响应和写响应）进行加密和认证。

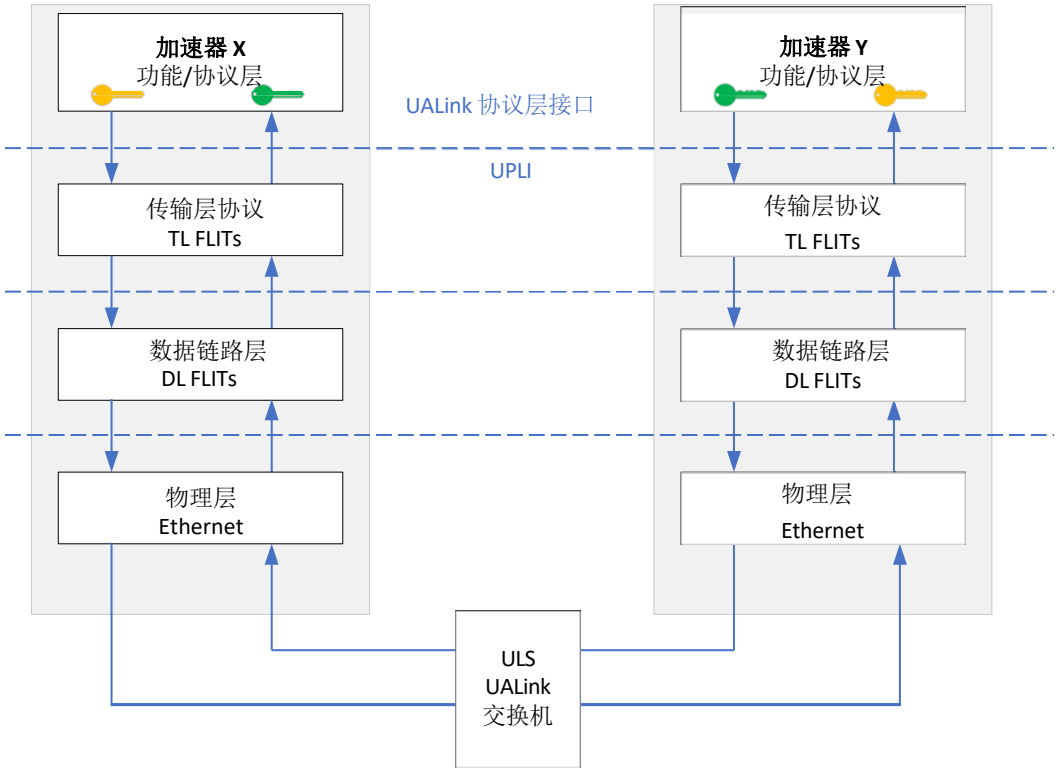

图9： 端到端加密和认证

5 可管理性

UALink POD 由一个或多个通过ULS连接的加速器组成，并由 POD 控制器进行管理。每个加速器都部署在 UALink 系统节点上，加速器间的数据流可以通过 ULS 进行路由。多个 UALink POD 可相互连接以构建更大规模的加速器集群。UALink 规范本身并未对 POD 间的通信与控制作出规定。

图 10 展示了一个 UALink POD 设置示例，包含交换机和服务器平台。图中显示了一个包含四个 UALink 加速器（每个有三个端口）、两个主机 CPU、一个 NIC 和一个 BMC 的系统节点。UALink 交换机负责在加速器之间路由，并由称为交换机管理代理的软件管理。物理交换机在硬件中实现，但出于路由和创建虚拟 POD 的目的，可以被划分为多个逻辑交换机。

通常情况下，物理交换机托管在交换机平台上，该平台在处理器（例如 x86 CPU 或BMC）上运行交换机管理代理。该处理器通过高速接口（如 PCIe®）连接到每个物理交换机，并连接到用于与 POD 控制器通信的网络接口。

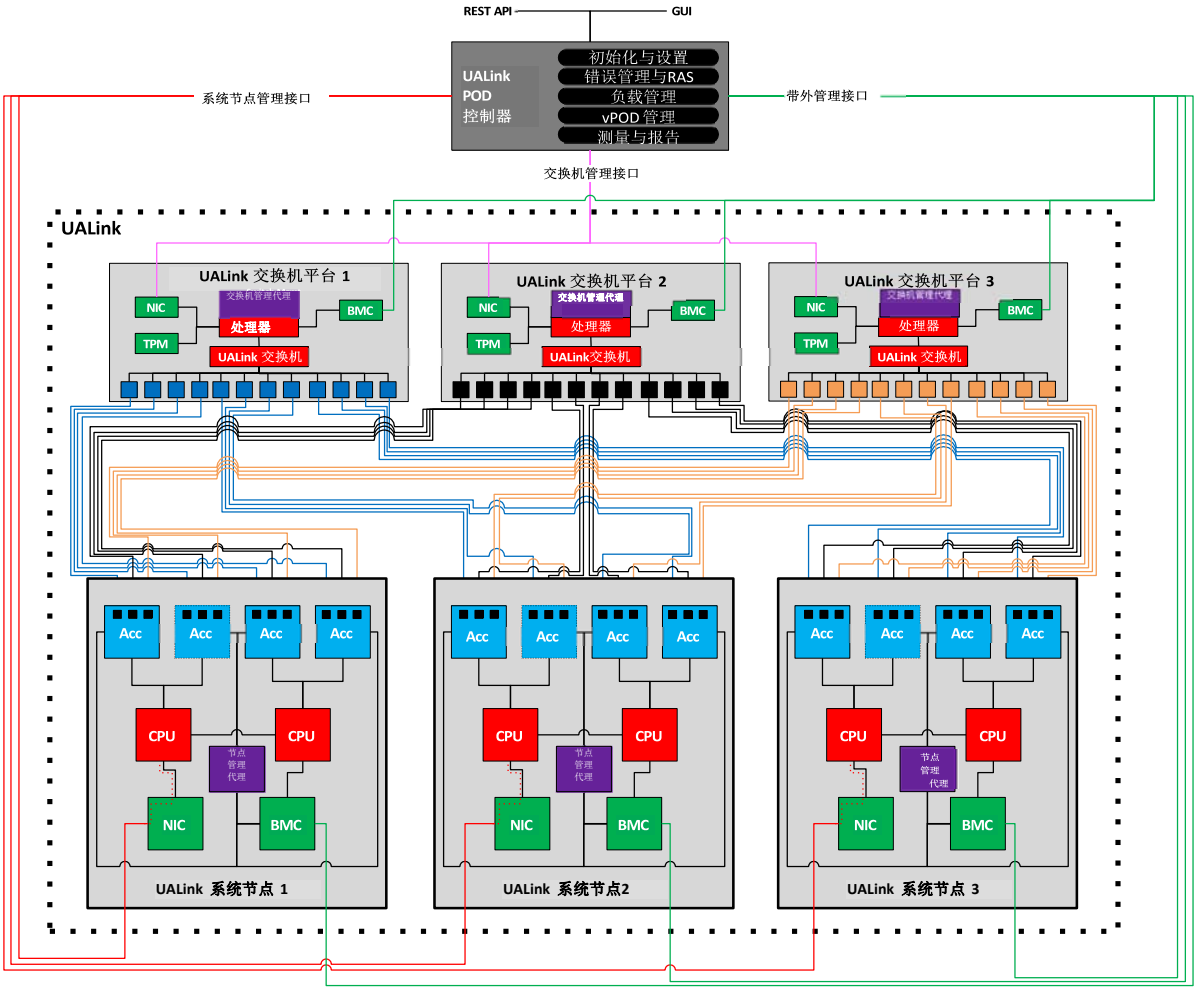

图10： UALink机架级系统管理接口

6 小结

UALink 是开放的、标准化的Scale-up互连技术，旨在变革数据中心的连接方式。UALink 是开启计算效率新时代的关键，专为同构的加速计算设备之间的超高带宽和低延迟数据交换而量身定制。

UALink 处于人工智能和机器学习领域创新的前沿，借助无处不在的以太网生态系统，为专用加速器互连提供了一条开放的生态系统路径。通过集成 UALink 交换机，具备 UALink 能力的加速器能够深度纵向扩展，创建超高带宽的多节点加速器 POD。UALink 还通过支持在多达 1024 个加速器组成的整个 POD 上进行Load / Store操作，实现了一种简单的软件模型。UALink技术将进一步提升未来的 AI 和 HPC 应用的性能和易用性，同时降低总拥有成本（TCO）。由 UALink 联盟开发和推广的这项技术将产生持久的影响。

关于Ultra Accelerator Link Consortium

Ultra Accelerator Link (UALink) Consortium 成立于 2024 年 10 月，是一个开放的行业标准组织，致力于开发 UALink 规范——一种适用于加速器Scale-up的高速互联技术，可提升下一代 AI 和 HPC 集群的性能。该联盟由行业领军企业组成的董事会牵头，包括阿里巴巴、AMD、Apple、Astera Labs、AWS、Cisco、Google、HPE、Intel、Meta、Microsoft 和 Synopsys。联盟制定的技术规范不仅能为新兴的人工智能应用模式带来突破性的性能提升，还将为数据中心加速器构建开放的生态系统提供支持。

有关 UALink 联盟的更多信息，请访问 www.UALinkConsortium.org

有兴趣加入吗?

即刻加入 UALink 联盟，参与技术工作组并影响 UALink 规范的发展方向。在 此处 了解有关 UALink 会员的更多相关信息。

ULTRA ACCELERATOR LINK™