中国移动
China Mobile

开放云网络之高性能网关技术白皮书

开放云网络助力多云互联

WHITE PAPER ON HIGH-PERFORMANCE GATEWAYS FOR OPEN CLOUD NETWORKS

2023

中国移动通信集团有限公司

前 言

云网络（也称作 SDN）是实现公有云环境下多租户隔离的关键技术手段，而云网络网关（也称作 SDN 网关）作为云网络系统的关键转发网元，在云网络系统中扮演着至关重要的角色（注：在本技术白皮书中，“SDN”和“云网络”是可以相互替换的术语）。本技术白皮书对当前业界主流的云网络网关技术路线进行了较为全面、客观的优劣势分析，并结合对云网络网关相关技术的现状和趋势的研判，提出了将有状态网关和无状态网关解藕并分别采取不同技术路线实现高性能网关的技术思路。该技术思路不仅符合开放网络的理念，同时也顺应极简网络的潮流。目前移动云正携手合作伙伴推动上述技术路线的落地部署。通过该技术白皮书的技术分享，希望在业界形成更广泛的共识，共同推动云网络朝着更加开放的极简网络的方向演进。

编写单位和人员

中国移动：姚军，王燕，徐小虎，徐硕，俞文俊，谭跃辉，李维亮，徐璐，金鹏程，主要负责技术白皮书整体章节的构思以及第一章、第二章以及 4.1 章节的编写以及所有章节的修订。

北京邮电大学：潘恬，主要负责第二章的编写

英特尔：陈志华，主要负责 3.1 和 4.2 章节的编写

博通：张玺，何宗应，王娜，曲延光，主要负责 3.2 章节的编写

锐捷：吴航，邹赛虎，章建钦，主要负责 4.3 章节的编写

新华三：王雪，主要负责 4.3 章节的编写

目录

第 1 章　云网络网关功能定位及分类

云网络（也称作 SDN）是实现公有云环境下多租户隔离的关键技术手段，而云网络网关（也称作 SDN 网关）作为云网络系统的关键网元，在云网络系统中扮演着至关重要的角色，如实现租户流量集中转发控制(比如跨 VPC 的互访)和复杂业务处理逻辑（比如网络地址翻译即 NAT 和服务器负载均衡即 SLB）。

按照是否需要维护租户流量的会话状态（比如 TCP/UDP 会话），可以将云网络网关分为无状态网关和有状态网关两类。顾名思义，无状态网关无需维护会话状态，而有状态网关则需维护会话状态。

1.1　无状态网关

无状态网关通常只需要维护路由转发表和静态的地址映射表。VPC 网关（以下简称 VGW）和互联网网关（以下简称 IGW）是典型的无状态网关。VGW 主要维护 VPC 路由信息，IGW 除了维护 VPC 路由之外，通常还需要维护 VPC 私网地址和公网地址的 1:1 NAT 映射表和功能用于 1:1 NAT 功能。IGW 的 1:1 NAT 功能有别于下文中提到的 NAT 网关的功能，1:1 NAT 主要完成 VPC 地址到公网地址的 1:1 翻译，1:1 NAT 映射条目在租户购买公网 IP 地址之后就通过

SDN 集中控制器配置生成并下发到 IGW，而不是由途径 IGW 的 TCP/UDP 等流量触发创建，因此，IGW 的 1:1 NAT 条目数量取决于 IGW 所在的云数据中心对外已售卖的 EIP 地址数量，而与会话数量无关。

1.2 有状态网关

有状态网关除了需要维护少量条目的路由转发表之外，需要动态创建并维护由途径的 TCP/UDP 等流量触发的会话连接表条目。NAT 网关和 SLB 网关是典型的有状态网关。NAT 网关主要完成 VPC 内主机主动访问互联网所需要的 n:1 SNAT 功能，即源地址翻译功能，由于某个 VPC 内的多个 VPC 私有地址会复用一个关联到 NAT 网关的公网地址或 NAT 网关的私网地址（注：该私网地址将在 IGW 上完成 1:1 NAT，将该私网地址转化为公网地址）对外通信，NAT 网关需要维护 TCP/UDP 会话连接状态来实现多种用途，比如保证一个内部主机主动发起的任意一个 TCP/UDP 会话的所有数据包在经过 NAT 网关时，源地址和源端口翻译之后的源地址和源端口始终保持不变。SLB 网关主要完成四层负载均衡能力，不论是采用 Full NAT 模式，DR 模式还是 DSR 模式，SLB 网关都需要跟踪维护会话连接状态信息，以保证同一 TCP/UDP 会话的流量始终被负载分担到服务器集群中某个固定的真实服务器 Real Server 或 L7 负载均衡服务器（比如 Ngnix 服务器）。

此外，VPN 网关也属于有状态网关，因为其需要与对端 VPN 设备（如 VPN 客户端）建立和维护 IPsec 或 SSL 会话连接。

第 2 章　当前云网络网关主要技术路线

针对上述提到的云网络网关（包括有状态网关和无状态网关）的技术实现路线，不同云厂商采用的技术路线往往不尽相同，比如，有的完全采用 NFV 方式实现有状态网关和无状态网关，有的则采用超融合硬件设备来实现有状态网关和无状态网关，有的则采用 NFV 方式实现有状态网关而采用可编程硬件设备实现无状态网关。

2.1　NFV 形态网关技术路线

云网络发展早期主要使用厂商硬件设备方案支撑相关网关业务发展，但随着云网络业务的快速发展，上述方案暴露出的问题越来越明显，比如设备的采购成本和维护成本高昂，设备的规格和特性无法快速升级迭代，已经严重制约了云网络的快速和可持续发展。由此，云网络网关开始往 NFV 网关技术方向演进，相关技术经过多年的发展，目前已经经过了三轮迭代演进，分别是 NFV 1.0、NFV2.0 和 NFV3.0。

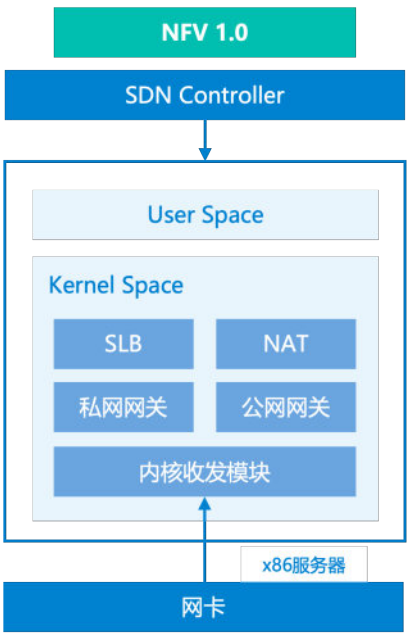

图 2-1 NFV 1.0 方案

基于 Linux 内核的 NFV1.0 方案，如图 2-1 所示。在 Linux 内核中，基于网络子系统的 Netfilter 模块，开源组织开发出了用于负载均衡的 LVS 和用于防火墙、NAT 的 Iptables。云厂商将这部分功能进行整理和自定义，完成了 NFV 1.0 的方案。该方案基于开源系统，可以在有新需求的时候进行快速开发迭代，并且该方案运行在 x86 服务器上，当设备性能和容量不够时可以方便快速地扩容 x86 服务器集群。NFV1.0 方案很好地满足了云网络业务快速开发迭代、快速扩容的需求，但是受限于 Linux 内核相对复杂冗长的报文处理逻辑，单个 NFV 网元设备的性能始终无法达到较高水平。

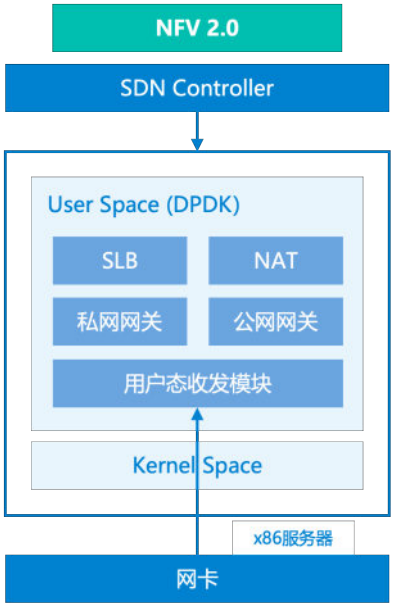

图 2-2 NFV 2.0 方案

随着 DPDK（Data Plane Development Kit，数据平面开发套件）技术的出现和不断成熟，业界开始使用 DPDK 技术开发用户态的网关系统，由此从 NFV 1.0 进入到 NFV 2.0 阶段。在如图 2-2 所示的 NFV 2.0 方案下，通过内核旁路、核隔离、独占网卡、内存巨页、网卡多队列和 DPDK 用户态协议栈等一系列优化技术的使用，单个 NFV 网元的性能得到了大幅提升，已经可以达到 10/25G 网卡线速转发能力，相比内核态网关，大约有 10 倍以上的提升。

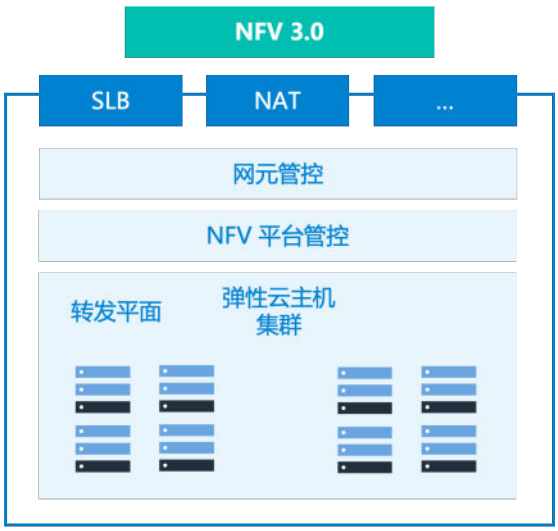

图 2-3 NFV 3.0 方案

虽然在 NFV 2.0 阶段解决了性能问题，但随着云网络的进一步发展，弹性能力不足、开放性不够等新的问题又开始凸显出来。为进一步解决弹性扩容、支撑第三方网元等问题，业界开始了 NFV 3.0 的演进。NFV 3.0 方案如图 2-3 所示。在 NFV 3.0 方案中 NFV 网元不再部署在物理服务器上，而是使用通用云主机（即虚拟机），这样网元便拥有了普通云主机的极致灵活性，具有极强的弹性伸缩能力，也不再需要适配新的网卡硬件。网元开始被统一的 NFV 平台纳管调度，这就使得其开放性大幅提高，云上用户自己的第三方网元也可以运行到 NFV 平台上，被 NFV 平台调度，提供服务给云上其他租户使用。

基于 NFV 形态的 SDN 网元拥有了极致的弹性和开放性，这极大促进了云网络网关的快速发展。然而随着云网络的持续快速发展，在一些场景下开始出现一些 NFV 方案无法解决的问题，主要体现在以下两方面：

1、无法处理超大单流

NFV 通过进行横向弹性扩容来支持大流量的处理，但这种大流量有个前提，即整体流量中流比较均匀且每条流都不能太大。如果流量中出现了一些超大的单流或者 HASH 分发不均匀，则 NFV 方案将遇到性能瓶颈，因为 NFV 方案中网络流量转发依赖于 CPU 核的软件转发，而单个 CPU 核的转发能力是存在一个上限的。当单个 CPU 核上承载的流量超过其上限时将会出现过载丢包，影响大单流的租户和该 CPU 核上的其他租户。

2、超大规模部署下的高昂成本

NFV 的横向弹性扩容将会使得在处理超大带宽时引入超大规模的网元集群。假设某个集群达到了 1Tbps 的带宽，以单个虚拟网元 8Gbps 处理能力计算，则需要至少 125 台虚拟网元，如果考虑冗余容错等其他因素，则需要更多的虚拟网元。这将导致 NFV 集群的整体拥有成本即 TCO 变得十分高昂。

2.2 超融合硬件形态云网络网关技术路线

近年来，一些云厂商采用了超融合硬件网关来实现高性能 SDN 网关功能。超融合硬件网关也称作 Server Switch，其包含了 Tofino、x86 CPU 甚至 FPGA 等硬件资源，将所有网元包括有状态网关和无状态网关统一在超融合硬件网关内实现。

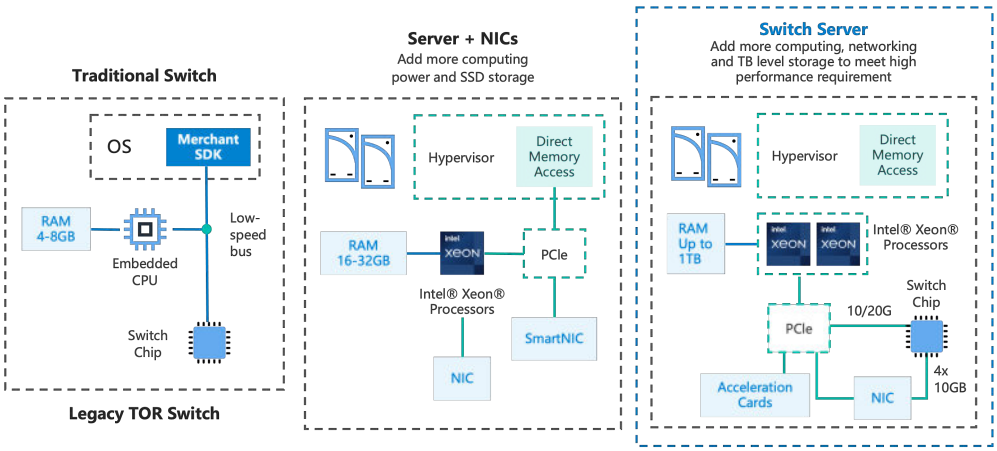

图 2-4 融合 Tofino、x86、FPGA 的硬件设备

当前 Tofino 芯片的片上内存容量（SRAM、TCAM）相对较小，无法大规模云网络对数以百万计的 VPC 路由容量需求。为此，需要从多个维度对表项资源的分配使用进行优化：第一个维度是使用多级存储转发架构，用 FPGA 甚至 CPU 作为可编程交换机中可编程交换芯片有限片上内存资源的补充，可编程芯片没有流表命中的流量将转到 FPGA 甚至 CPU 处理；第二个维度是使用多台可编程交换机实现表项的水平分割；第三个维度是在单台可编程交换机上，使用流水线折叠等技巧压缩表项的存储空间占用（如图 2-5 所示）；第四个维度是基于可编程交换机集群和 x86 服务器集群的两级转发处理架构，其中流量经过负载均衡器之后首先进入可编程交换机集群进行快速处理，因为可编程交换机交换芯片的片上内存容量有限，没有流表命中的流量将转到 x86 服务器集群处理。

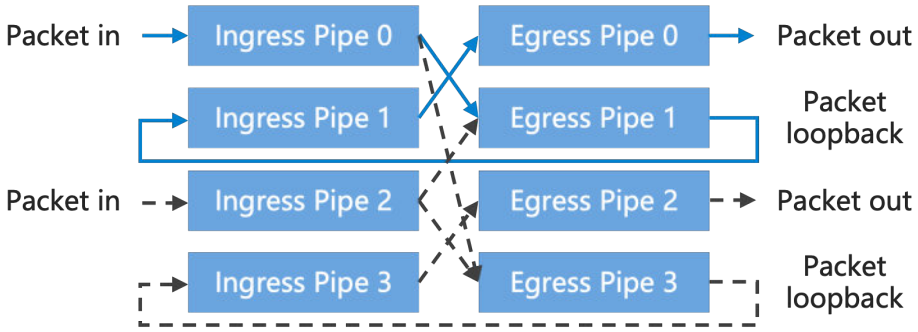

图 2-5 基于流水线折叠优化 Tofino 芯片的存储占用

超融合硬件形态网关的流量转发模型参考如下图 2-6 所示：

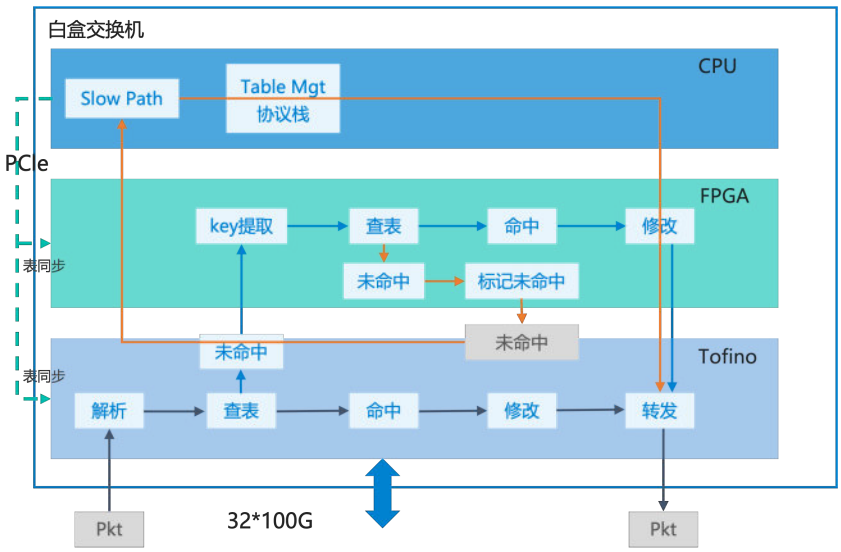

图 2-6 超融合硬件网关的流量转发模型参考

首先，报文在 Tofino 查表命中会进行快速转发；然后，若报文未命中 Tofino 表，会上送到 FPGA 查表命中后转发；最后，如果报文均未命中 Tofino 和 FPGA，会上送 CPU 后进行转发。

超融合硬件网关优势主要体现在超强的包转发处理性能，例如单个 Tofino 芯片就可以提供 12.8Tbps 的转发性能，相当于几十台的 x86 服务器的性能，且由于单个流水线转发吞吐极大，流量突发或汇聚产生的网关打爆导致丢包现

象将很少出现。

超融合硬件网关同样存在以下两方面的不足：首先，超融合硬件网关将 CPU、DPU、FPGA 和可编程芯片（如 Tofino）集成在一个硬件设备上，且需要上述转发单元之间的转发逻辑协同，系统架构相对复杂，开发和维护技术门槛较高；其次，超融合硬件网关通常基于领域特定芯片提供极致的性能，这类芯片产量较少，盈利模式不稳定，因此其技术路线的发展也同样容易出现变数，例如 Intel 突然宣布不再支持下一代 Tofino 芯片的研发，这给云厂商技术路线的选择带来挑战。

第 3 章　云网络网关相关技术现状和趋势

3.1　DPU/IPU

根据统计，数据中心的网络带宽复合增长率从 10 年前的 30%，已经增加到近年的 45%。这些网络任务对计算能力的需求在不断增长，采用传统的主机 CPU 处理的方式已经不堪重负。一方面为了减轻主机 CPU 的负载，另一方面提高网络吞吐的性能，数据中心的网络接口在不断的升级换代。从普通的传统网卡，到带硬件卸载的智能网卡，今天已经来到了 DPU（数据处理器）/IPU（基础设施处理器）的时代。

DPU/IPU 作为一种新型的数据处理器，可以满足网络、存储和安全的硬件卸载和加速。针对 DPU/IPU 的设计，其实国内外有很多不同的架构，比如 ASIC+CPU、FPGA+CPU、NP+CPU 等等。由于 ASIC+CPU 在大规模部署上有成本优势，且它通过 ASIC 可以提供强大的硬件卸载功能、通过 CPU 提供灵活的计算功能，ASIC+CPU 架构将可能成为市场的主流方向。

DPU/IPU 作为网络处理的接口，它首要解决的问题是带宽的吞吐。因此，25G、100G 接口是基本的要求，而 200G、400G 的更高速率接口也已经开始

商用。但是，仅仅具备网络接口还只能充当普通的网卡，还远远不能满足数据中心网络的要求。因此，DPU/IPU 还需要提供硬件卸载、加速的功能，才能释放其高达 400G 的网络吞吐性能：

- 支持主机或虚拟机使用高性能 SR-IOV 和传统虚拟化 VirtIO
- OVS/OVS-DPDK 将数据平面和控制平面都能够卸载到 DPU 上，主机上无须运行这些软件，整个系统成为一个安全隔离和精简的平台
- 支持 RDMA/RoCE（远程内存直接访问），大幅度降低内存读写时延
- 专用芯片针对 IPsec、TLS、MACsec 等进行硬件卸载来提升性能。

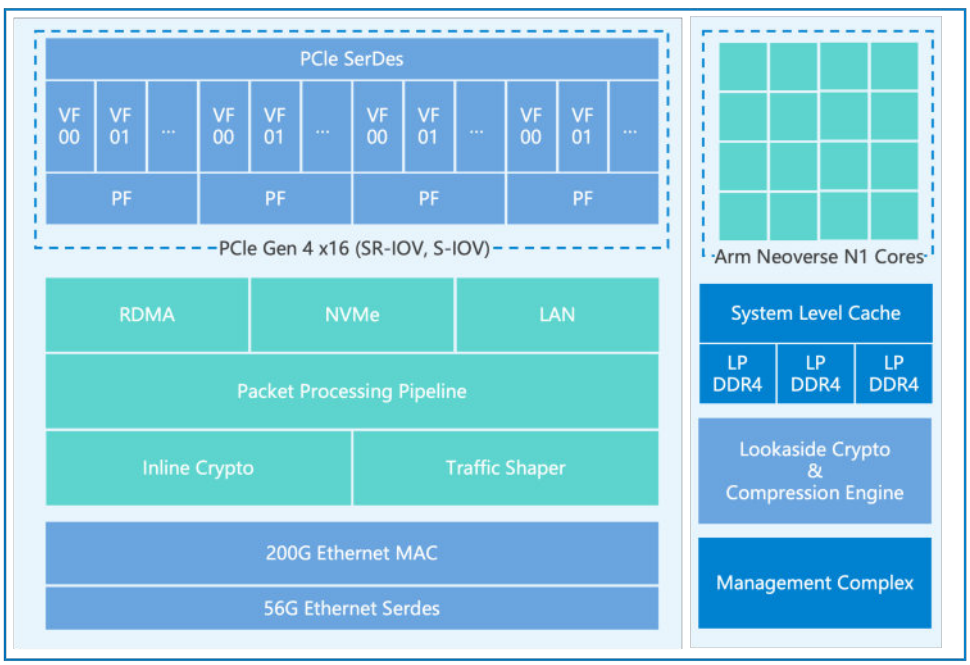

图 3-1 Intel IPU ES2000 系统结构示意图

ES2000 是 Intel 最新一代 IPU 产品。如上图所示，ES2000 具有 800M PPS 处理能力的硬件 pipeline，支持多次 recirculation。另外，ES2000 包含 3 个

LPDDR4 控制器，最大可支持 48GB DDR，报文处理所需的所有表项，包括 ACL 表，路由表，各种查找表，连接跟踪表等，都可以直接存储在该 DDR 中，无需借助 Host Memory。正是如此，ES2000 可在支持千万级别的并发连接情况下，仍然能够支持 200Gbps 线速的双向处理。此外，ES2000 包含一个 200Gbps 双向线速处理能力的 Inline IPSec 引擎。由此可见，ES2000 可以满足高性能有状态 SDN 网关的相关指标要求。

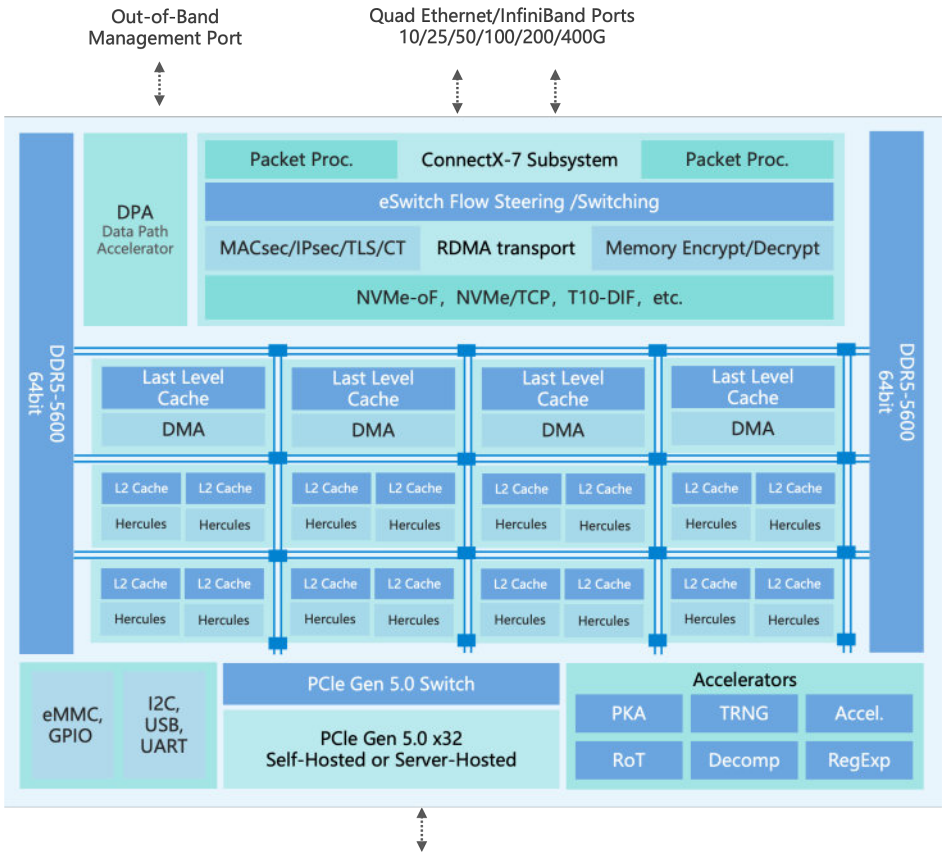

图 3-2 英伟达 DPU BF3 系统结构示意图

如上图的英伟达 BlueField-3 产品（采用了 CX7 AISC 和 ARM A78 Hercules CPU core）是 CPU+ASIC 架构的 DPU 的典型代表。这个架构采用了内置的 PCIe Switch，方便与主机、其它模块、甚至与多 DPU 的灵活互连，而

且互连带宽能够得到保障，也是 DPU 业界的发展趋势之一。

产品	BlueField-2 DPU	BlueField-3 DPU
带宽	200Gbit/s	400Gbit/s
DPDK 最大包转发速率	215Mpps	280Mpps
RDMA 最大包转发速率	215Mpps	370Mpps
计算能力	SPECINT2K17:9.8	SPECINT2K17: 42
内存带宽	17GB/s	80GB/s
VirtIO 加速	40Mpps	92Mpps
VintIO 延迟	16μs	14μs
每秒连接数(Connection Per Second，CPS)	1.5M	8M
IPsec 加速	100Gbit/s	400Gbit/s
TLS 加速	200Gbit/s	400Gbit/s
MACsec 加速		400Gbit/s
NVMe SNAP	5.4MIOPS @4K	10M IOPS @4K
NVMe/TCP	2.1M IOPS	5M IOPS

图 3-3 BF2 与 BF3 关键性能指标对比

从上图中 BlueField-3 与其前一代产品即 BlueField-2 的硬件卸载性能对比数据可以看出，不管是接口带宽、加密性能、有状态的并发会话数量，BlueField-3 相比前一代产品都有了大幅提升，BlueField-3 完全可以满足高性能有状态 SDN 网关的相关指标要求。

3.2 新一代可编程网络芯片

为了满足网络未来可见的发展，同时准备适应潜在的变化，网络数据面不仅在容量和性能上要保持超过摩尔定律的发展速度，在灵活性上也需要进一步提高。网络数据面的灵活性的一个重要体现就是网络数据面的可编程能力。

上一代的高速大容量网络芯片的灵活性体现在如下层面：

1）芯片转发层面

芯片厂商通常实现并提供标准的数据面转发功能。对于网关等特定应用场景的个性化转发需求，用户可以使用特定的芯片转发层面可编程语言，自行开发可编程应用程序，定制化报文处理以及转发的流水线。可编程应用程序通过芯片厂商提供的工具链生成二进制镜像文件并由 SDK 加载到芯片以实现定制化转发。

2）SDK 层面

运行在主机 CPU 上的控制面通过 SDK API 调用传递信息到“片上资源管理逻辑”，进而通过内部通道来配置不同的片上逻辑，到达相应的目的。

在此之上，为了能够适配更多的应用场景，并提升编程效率，新一代的可编程芯片通常如下特性：

- 基于流水线逻辑的可编程
- 基于表项资源的可编程
- 基于队列调度的可编程
- 基于加速引擎的模块化设计
- 更加友好的可编程软件

3.2.1 基于流水线逻辑的可编程

作为可编程芯片的核心功能，基于流水线逻辑的可编程已经广泛被上一代

可编程芯片所支持。

一个数据包的处理，至少要涉及端口，入方向处理，查找表资源，数据包缓存和出方向处理。数据面在这几个环节的灵活性（可编程）至关重要。

“入方向处理”部分，数据包包头的解析为后继动作提供重要输入，逻辑上可以表达为一个有向图，解析步骤的可定义，合理的解析深度是数据面可编程的重点之一。

在“入方向处理”部分，数据包头解析之后，要根据层次化的结果，通过查找表，确定后继处理（转发处理）的数据和处理逻辑，这个部分，广域网 PE 设备的要求最高。

转发处理的数据和处理逻辑确定之后，根据目的地址的转发处理是相对简单的，这里的挑战是查找表的设计与组织。这种挑战体现在三个方面：

1）芯片设计环节

网络芯片通常会提供多种不同的查找方法，包括严格匹配(exact match）、最长匹配查找（ALPM/LPM）、地址索引查找以及 TCAM 查找。查找表内容（包括 Key 以及结果）通常放在内存（SRAM 以及 TCAM）里面。芯片设计时候需要根据其策略确定查找表硬件资源的基本单元构成、支持的查找次数、内存大小以及布局方式。尽管不同芯片设计偏向不同，主流芯片通常使用并行查找、查找的 match key 与查找结果相分离、查找表资源跨 stage 局部或者全局共享、查找表存储和处理逻辑的分离、片下可扩展表项资源等技巧以达到在可编程硬件

资源最大化的前提下能够实现功耗和成本的平衡。

2）可编程应用程序开发环节

网关数据平面的开发者需要根据自己的业务诉求设计并定义各张逻辑表以及相应的转发流水线。不同的设计方法，虽然可以达到相同的系统功能，但可能逻辑表容量可能会有极大影响。所以逻辑表以及流水线设计需要非常仔细。同时为了尽量获得更大的表容量，在可编程芯片开发工具链的帮助下，需要对逻辑表以及流水线设计进一步调优。

3）可编程芯片开发工具链

通常由芯片厂商提供，用以将前述可编程应用程序高效地映射到芯片的物理资源。工具链效能会极大影响开发者可见的逻辑表提供能力以及物理资源利用效率。

“入方向处理”的后部分，是访问控制列表处理和 QOS 处理，访问控制列表处理结果可以影响转发结果，或者为后续的 QOS 处理提供数据流的染色；访问控制列表一般采用精确匹配或者掩码匹配，访问控制列表查找键值的组成灵活性，宽度和逻辑操作能力是数据面可编程的另外一个重点，这里的可编程并不是指程序逻辑的多样性，而是程序入数据的宽度/广度。

而“出方向处理”和“入方向处理”相比，减少了转发处理部分，增加了数据包头的编辑部分。

“出方向处理”数据包头的解析是网关功能的一个重要支持基础，例如，

NAT 功能一般安排在入方向处理完成，出方向的一些功能可能需要根据 NAT 后的新的包头来处理，这样“出方向处理”数据包头的解析就是一项必不可少的步骤。

“出方向处理”数据包头的编辑能力是数据面灵活性又一重点，前面的处理步骤产生了众多的状态和结果，根据这些状态和结果，生成相应的新的数据包头，体现了数据面在一个处理节点上的处理结果，同时又为数据面的下一个节点提供新的信息。

上述对“入方向处理”和“出方向处理”的简而又简的描述，可以看出目前数据面可编程的着重点在数据包解析，转发查找和数据包编辑。

3.2.2 基于表项资源的可编程

虽然基于流水线的可编程极大的增强了网络转发芯片的灵活性，但是实际的开发和部署中，芯片表项资源数量的限制成为了很多新型应用的性能瓶颈。比如在数据中心网关应用中，通常会需要超大规模的路由表，NAT 表以及 VxLAN 封装/解封装表，而传统的可编程网络转发芯片对这种场景的支持有限。

商业网络转发芯片会内置一定规模的表项存储资源用于表项查找，如 SRAM 和 TCAM。然而，上一代的可编程网络芯片通常存在如下问题：

1）资源规模较小

市面上常见的网络转发芯片的内置资源规模通常只能支持万级或十万级的

路由规模，无法支撑现代数据中心中网关，移动网数据面功能等海量用户场景。近年来，业界已经关注到这个问题并着手解决，如博通公司的量产可编程网络转发芯片的内置资源已经可以支撑两百万以上的路由表，并计划在下一代可编程网络转发芯片中继续提升内置表项的容量。

2）资源的分配的灵活性较差

在现有量产的可编程交换芯片中，一个硬件存储资源块区只能被分配到一个“Stage”中，所以用户需要对这些硬件资源进行非常仔细的管理和规划。

在新一代的网络转发芯片中，芯片内部的存储资源通常被抽象为一个资源池，并可以被所有的“Stage”引用，而不再需要考虑和设置资源和“Stage”之间的映射关系，极大的提高了全局资源利用率。

3）芯片资源的定义和流水线设计的耦合性较强

如上所说，当一个功能表项需要的资源数量过多时，需要在设计流水线时进行特殊的设计，如采用多次内容相同的查找来实现海量表项的功能。而这给芯片流水线设计的工程师带来了很大的困难，因为工程师可能需要定义多种不同的流水线结构来应对不同的应用场景。

新一代的可编程交换芯片为了解决这个问题，将资源的可编程和流水线的可编程彻底解耦。用户可以灵活的定义芯片内部资源的分配情况，并可以独立于流水线设计进行内部资源分配模板的加载。这样带来的好处是针对不同应用场景的同类型产品，使用一套流水线模板和多套芯片资源配置模板即可完成对设

备的定制。比如，一个网关产品对于 NAT 需求较为强烈的场景和对路由需求较为强烈的场景，只需要准备两个不同的资源配置模板即可，而流水线模板不需要做任何变化。

3.2.3 基于队列调度的可编程

以往的可编程网络转发芯片主要覆盖了数据面的流水线处理，但是对于 MMU 和流量调度的可编程则支持非常有限。

比如对于 MMU，并不在传统的“可编程”范畴之内 – 流水线的开发工程师只需要关注在入/出流水线的逻辑表项设计和资源配置即可，而 MMU 只是作为一个报文缓存和转发的单元被独立在可编程视图之外，而这带来了整个可编程逻辑的割裂。

以博通的 TD4 为代表的新一代的可编程网络转发芯片，已经把 MMU 的抽象成一个加速引擎，并将其引入到整个可编程的视图之中。这样，开发工程师可以非常清楚的确认在连接到 MMU 的总线上，什么信号需要被入流水线送到 MMU，以及出流水线需要得到什么样的信号。

除此之外，对于很多运营商应用来说，网络转发设备需要针对不同用户等级提供对应的 SLA 保障服务，而这对网络转发设备提出了如下要求：

- 队列数量的要求
- 调度层次以及调度模式的灵活配置

- 海量的计数器、流量整形器资源，并可灵活挂载到不同的用户流量上

传统的网络转发芯片通常只支持非常有限的 HQoS 调度层级，并且调度模式固化，无法通过编程的方式来自由和灵活的定义 HQoS 的调度模型。

新型的可编程网络转发芯片，比如博通的 StrataDNX 系列，则同样也支持流量调度的可编程，包括可以允许用户灵活配置 HQoS 的层数，调度树状图以及灵活的限速等功能。

3.2.4 基于加速引擎的模块化设计

在传统的可编程网络转发芯片中，虽然提供给了用户能够自行定义流水线的能力，但是复杂度却较高。很多时候，用户需要通过 ALU 配合流水线设计来实现很多额外的操作，比如，序列号的产生和记录，从一个流中的多层报文头中提取最终的 QoS 信息，或者当目的地是 ECMP 或者 DLB 时的处理等，都需要用户进行繁琐的设计进行实现。上述设计极大的增加了开发工程师的工作量，并且会使得流水线设计代码冗长且降低可读性。同时，当芯片的可编程架构或编程语言出现升级时，也需要对这些功能视情况进行重新设计。

在新一代的可编程网络转发芯片中，一个明显的趋势是将这些常用的功能在硬件层面抽象成一个加速器模块，并且允许在可编程设计中通过可编程语言，在需要调用的流水线处直接调用这个加速器模块进行处理。这样带来的好处是，芯片的通用逻辑被抽象和屏蔽掉了，而开发工程师不再需要处理这部分内容；并

且由于业界主流的网络转发芯片厂商已经具有了丰富的部署经验，调用这些预定义的硬件加速器的稳定性会更好，并且效率会更为高效。

以博通的 StrataXGS TD4 芯片为例，该芯片已经集成了几十个预置的加速引擎，可以被应用到入流水线，出流水线，以及一些和 MMU，Counter 关联的各个位置。从功能角度讲，这些加速引擎包涵盖了正确性检查，QoS，聚合，ECMP，DLB，MMU，大象流处理等多种常用功能。用户在设计流水线时，只需要通过 NPL 调用对应的加速引擎库函数,就可以简单而快速的实现这些功能。

3.2.5 更加友好的可编程软件

网络数据面的编程与普通 CPU 应用程序的编程是截然不同的：

- 网络面的编程往往是覆盖数据包从入到出的全过程。
- 网络面的编程是数据流驱动的，而不是控制流驱动的。
- 网络面编程的一个典型特性是表项查找，查找表的生成是控制面预先生成的，一定程度上网络面编程和控制面编程是紧密耦合的。

除了使用常用的 C++语言之外，网络数据面应用程序通常使用为网络数据面编程定制开发的编程语言以更好地描述逻辑表和报文处理转发流水线，尤其重要地是如何更高效地把网络芯片的硬件能力和开发者的易用性完美结合起来。为了到达上述目的，网络可编程语言通常需要具备如下特点：

- 高效、简洁和易于使用

- 丰富的逻辑表定义能力
- 函数原语定义能力
- 支持并行查找
- 同一个报文命中多次查找时的冲突处理
- 报文的可视化能力
- 运行时的可编程能力
- 以简单地方式支持与控制面的高效集成
- 多种硬件芯片的支持能力

目前业内主流的网络可编程语言包括：

- NPL (Network Programming Language, https://nplang.org/)
- P4（https://opennetworking.org/p4/）

网络数据面可编程开发工具链里，编译器无疑处于技术核心位置，但仍需要其它工具辅助，以共建一套如下图的环境。

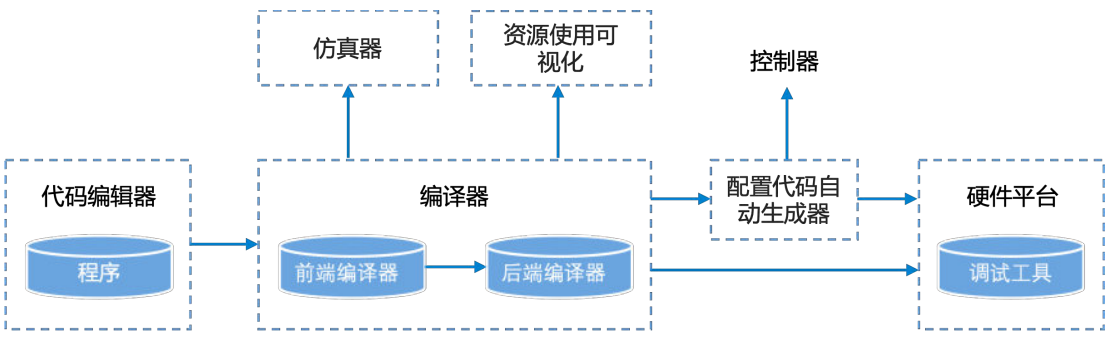

图 3-4 可编程芯片开发工具链参考

编译器通常分为处理语言、与目标平台无关的前端编译器和与特定硬件能力绑定的后端编译器。

对于前端编译器的设计，可以借鉴成熟的 C/C++等通用编译器，此外，鉴于减少在硬件平台上出错几率的考虑，通常需要提供对仿真器的支持。与特定硬件架构耦合的后端编译器的设计，除了编译速度、编译选项等这些通常的考量点外，还需要覆盖以下：

1）逻辑表的查找关键字和内容到物理表的映射与优化

可编程网关使用的查找表通常分为精确匹配表、掩码匹配表和索引匹配表，因考虑成本、功耗等因素，容量有限。编译器对物理表映射的处理方式决定了物理存储空间的使用效率。

2）报文解析、逻辑变量、逻辑运算、报文编辑等至物理资源的映射和优化

基于网络芯片的可编程网关，需要解析报文、通过查找表，指明报文路径、统计、整型并编辑报文。除了表项空间这个主要点之外，其余功能亦受资源大小限制。

3）对成熟通用功能的支持

尽管可编程网关因为场景不同，需求变化很大，但毕竟仍属于网络应用的一部分。已成熟的固定流水线交换芯片已经经过大规模实践应用，若将不需要变化的通用功能，提供类似于库函数的调用，即能节约成本，亦可减少出错概率。

4）资源约束检查

不同芯片有各自的架构和资源限制，如何提供基于特定平台的资源约束文件并进行检查，是后端编译器设计里很重要的一个步骤。

5）尽可能完善的错误提示信息

后端编译器使用前端编译器的生成文件，基本不可能如前端编译器那样提供准确的错误代码行信息，这也导致了后端编译的错误定位相对困难。因此，能否提供有效的错误定位信息，也是后端编译器设计的一个重要考虑点。

第 4 章　开放云网络高性能网关技术路线

4.1　有状态网关和无状态网关解藕

当前业界针对高性能 SDN 网关实现的技术路线，通常采用将有状态网关和无状态网关功能耦合的超融合网关设计思路，这类超融合网关不仅需要维护大量的路由表项，也需要维护会话创建的会话连接表项，比如将 NAT GW 的 n:1 NAT 功能集成到 IGW 上。但是，有状态网关和无状态网关的主要功能诉求不同，有状态网关和无状态网关紧耦合的超融合网关实现导致 SDN 系统的可扩展性差，要么是（NFV 形态网关）转发性能不足，要么是（多芯片硬件网关）系统架构复杂。

无状态网关（如 VGW，IGW）的核心诉求是高吞吐能力（如 Tbps 级别线速转发能力) 以及超大规模硬件表项规格能力(如 10K 以上的 VRF 规格，10K 以上的隧道规格，10M 以上主机路由表项规格)。因此，无状态网关的最佳技术路线是采用具有超大规模硬件表资源且高吞吐的可编程交换芯片实现上述两个核心诉求。由此带来的好处是系统架构简单，不存在多个转发逻辑单元（比如交换芯片即 ASIC 和 FPGA）之间的协同，系统开发维护技术门槛较低。此外，由于系统开放性好，集成第三方的具备高吞吐转发能力且具备大规模路由表能力的

商业路由器产品也具有一定的可行性。

有状态网关（如 NAT GW， SLB）的核心诉求是超大规模新建会话（如百万甚至千万级别）和并发会话（如千万甚至亿级别）处理能力，对网络转发逻辑的灵活性要求较高（如会话状态跟踪、会话限速），部分场景也存在高吞吐需求。最大新建会话能力是指每秒最大可处理的新建会话连接数，是个性能指标，这个指标与网关的计算性能密切相关。最大并发会话能力是指网关可以维持的最大会话连接数，是个压力指标，一旦会话数量达到这个最大限制，新的会话就无法被建立起来，这个指标与网关的内存大小有关。此外，由于转发性能加速的需求，主机内存中的并发会话连接表往往会被卸载到硬件转发芯片的 SRAM 中。有状态网关的最佳技术路线是采用 NFV 形态设备，通过扩展主机内存和增加 CPU 核数来解决大规模新建和并发会话处理需求，并通过 DPU/IPU 卡所具备的超大流表缓存能力，按需将流表信息加载到 DPU/IPU 以便进行硬件卸载，即将有数据转发需求的流表条目卸载到 IPU/DPU 的转发加速引擎，以此提升数据包转发性能。服务器是通用设备，DPU/IPU 通过标准的 PCIE 接口插槽加入服务器，整体系统开放解藕程度较高，可以实现快速迭代。

4.2 有状态网关的技术路线

为解决传统 NFV 形态的 SDN 网关转发性能瓶颈，可以采用 CPU + IPU 的硬件架构来实现新型的 SDN 网关。如下图所示，SDN 网关的各种核心应用运行在 CPU 上，通过慢速路径和快速路径的分离技术，将快速路径处理部分卸载

到 IPU 上来实现。利用 IPU 的 P4 可编程能力，可以根据上层业务需求，灵活的、有针对性的定制 IPU 上 Pipeline 转发行为，结合 IPU 千万级别的状态流表支持能力，在 IPU 上实现高性能、高复杂度、高定制化、有状态的快速转发。

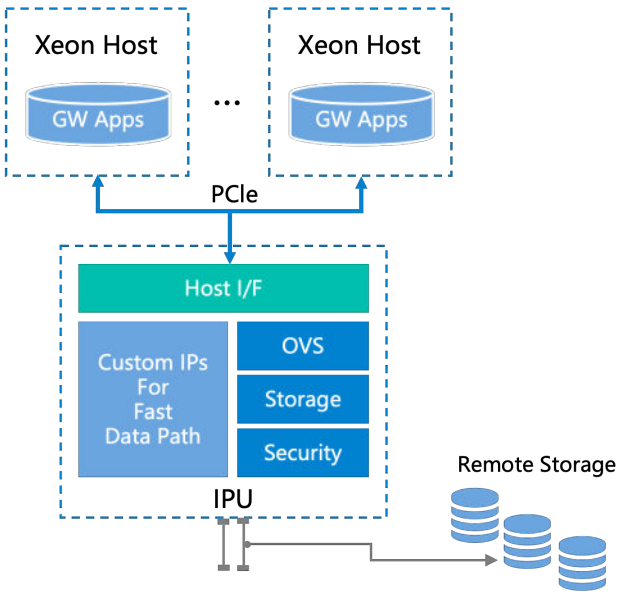

图 4-1 基于英特尔 CPU 和 IPU 的网关架构

从软件架构层面来讲，SDN 网关核心软件的底层由一组异构流水线控制软件构成，并面向网元业务功能做统一接口适配。网关核心软件的上层包括网络数据面软件和控制面软件。对于有状态的数据面软件，比如会话新建、流量均衡策略、连接跟踪等，提供模块化的网络功能参考设计，允许定制化模块的替换和重新组合。

以 4 层流量均衡(L4LB)为例，其快速匹配转发表可以由 P4 进行描述，然后把编译器生成的资源文件加载到目标系统的包处理流水线，如：IPU 的硬件 Pipeline。L4LB 网络功能模块对新会话、均衡算法及服务器选择等进行常规处

理，并将结果同步下沉到加速流水线上的快速匹配转发表，而后序的相同会话将由加速流水线进行高效转发。

Infrastructure Programmer Development Kit (IPDK)是一个开源的驱动和 API 框架（详细信息可参考 IPDK 网站：ipdk.io），针对基础设施提供提供卸载与管理，实现厂商和平台无关性。如下图所示，IPDK 包含多个功能组件，为网络虚拟化、存储虚拟化、Security、NFV 等多个应用场景提供软件框架支持。CPU+IPU 架构的 SDN 网关可以使用 IPDK 来构建底层的软件框架，一方面屏蔽底层硬件细节，实现硬件功能抽象化、跨平台化和高复用性；另一方面，对上层应用提供统一的 API 接口。

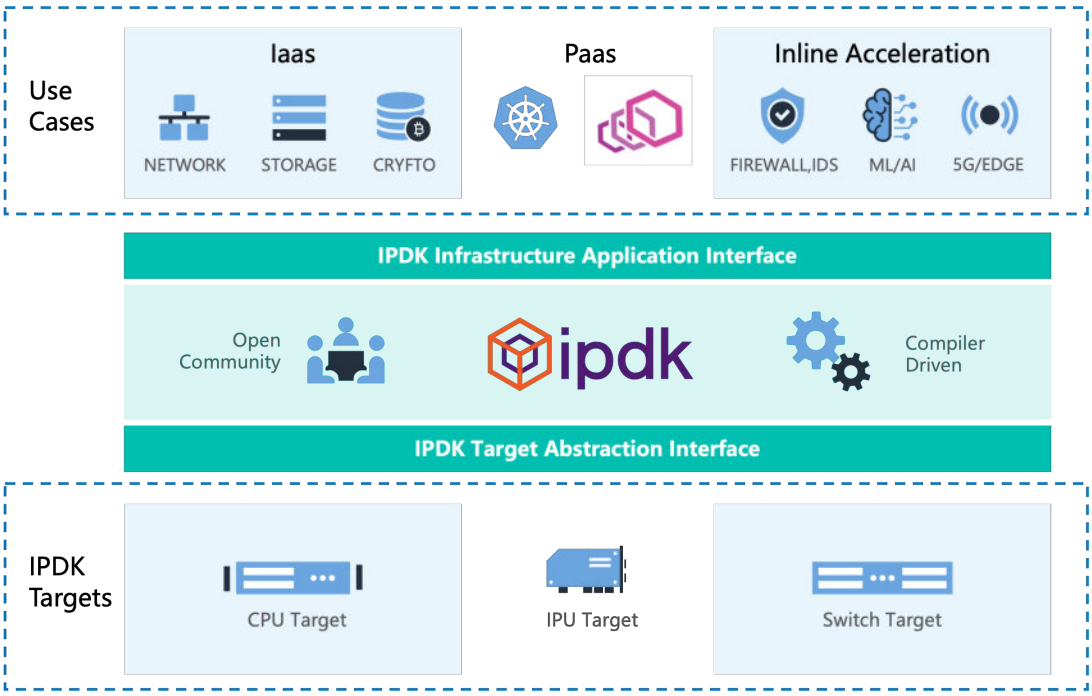

图 4-2 IPDK 架构参考

采用该架构的 SDN 网关，在延续 CPU 平台良好的通用性、扩展性和灵活

性的同时，还能借助 IPU 提供更高性能、更低时延的服务质量保障。

4.3 无状态网关的技术路线

网络可编程允许用户对网络数据面进行功能定义，构建按需定制、快速匹配个性化需求的网络，可以增强云服务商的网络自主可控能力，实现包括无状态网关在内网络功能。

自 2014 年 Barefoot（2019 年被 Intel 收购）推出 P4 网络编程语言之后，网络可编程技术得到了快速发展，在可编程网关、数据中心互联、带内遥测、自动化运维等方面已经有广泛实践和应用案例。

实现网络编程的技术基础是可编程芯片和网络编程语言，目前基本上大部分交换芯片厂商都号称他们的芯片支持可编程，但是从业界生态和技术成熟度来看，目前主要有因特尔的“P4 语言+Tofino 芯片”、思科的“P4 语言+Silicon One 芯片”、博通的“NPL 语言+TD/Jericho 系列芯片”三种技术路线。

因特尔的 Tofino 芯片系列

在 2019 年被因特尔收购之前，Barefoot Networks 是网络可编程领域的主要贡献者，其 2014 年发布 P4 可编程语言将网络推向了可编程时代。思科（Nexus 3400 系列）、Arista （7170 系列）等顶级网络供应商均推出基于可编程交换芯片的产品，谷歌、阿里、腾讯等大型互联网公司基于 P4+Tofino 架

构按需定制、快速匹配自身网络个性化需求，在数据中心互联、带内遥测、自动化运维等方面对网络数据面进行功能定义部署大规模部署可编程设备，从而增强网络的自主可控性。基于 P4+Tofino 芯片的网络编程是目前参与人员最多、经验最丰富生态最好的技术路线。

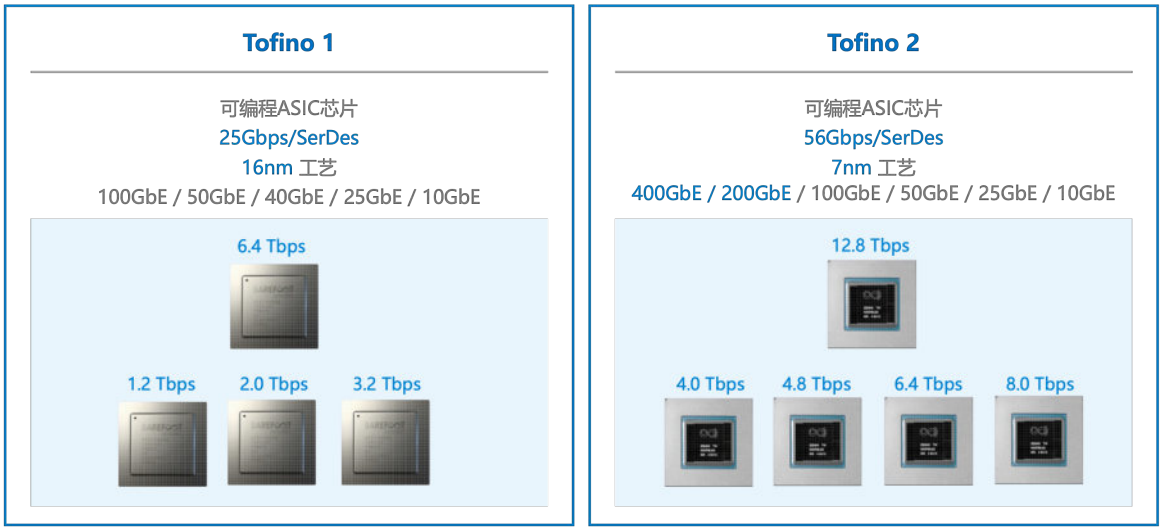

图 4-3 Tofino 1 和 Tofino 2 芯片家族

Barefoot Networks 的 Tofino 芯片规划了三代产品，2016 年推出 6.4T 的 Tofino 1，2018 年推出了 12.8T 的 Tofino 2，Barefoot 在 2019 年被因特尔收购之后 25.6T 的 Tofino 3 没有再发布，并且于 2023 年初因特尔对外宣布停止 Tofino 芯片的演进计划。因此，目前生态最好的 P4 + Tofino 技术路线陷入了无法继续演进的窘境。未来可编程技术会如何发展，该选择哪个技术路线作为未来的发现方向成为业内需要共同面对的一个问题。

思科的 Silicon One 芯片系列

思科于 2019 年对外发布 Silicon One 芯片。思科 SilconOne 系列芯片具

备大表项、大缓存及丰富的特性，Run to complete pipeline 架构优化报文转发，基于 P4 可编程语言（NPU），转发引擎可编程。面向园区网、运营商骨干网、数据中心网络等场景的高中低端路由器和交换机产品，提供完整的芯片解决方案，提供丰富的数据通信管理、内部和外部包缓存管理和 Telemetry 等功能。思科当前发布的产品中，既有框式设备（交换网、线卡），也有丰富的盒式产品，并进一步支持 SONiC 系统。目前思科的 SilconOne 芯片只对全球六大云服务提供商提供测试支持，P4 可编程方面暂未对外开发，仅可根据需求进行定制开发。对于研发无状态网关产品来说，需提需求，通过思科支持实现。

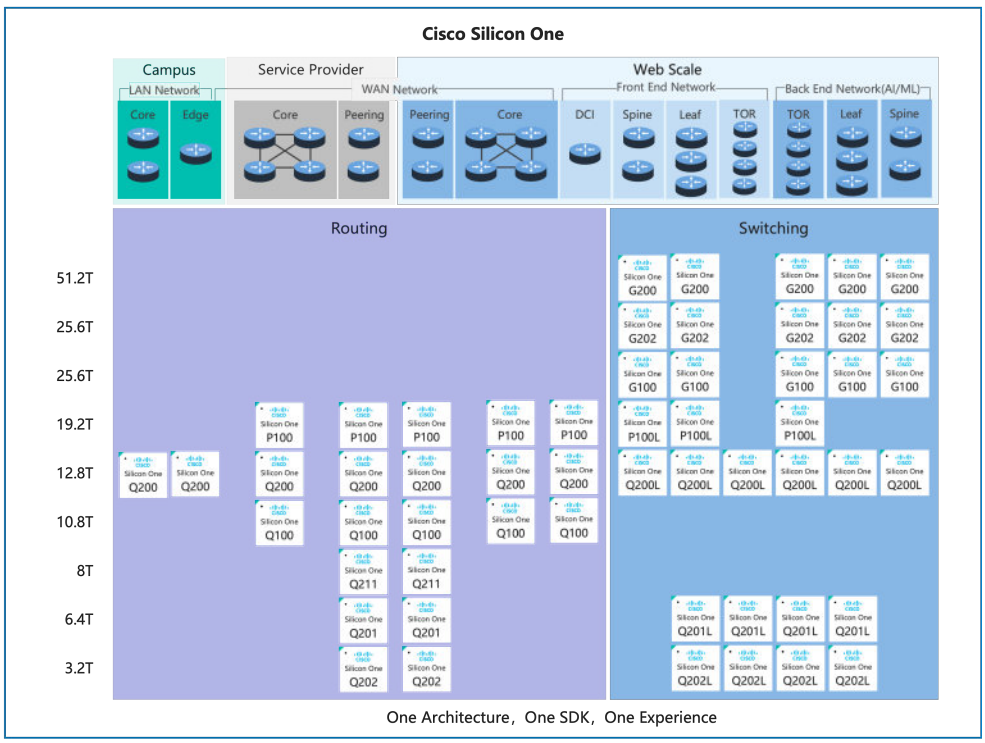

图 4-4 Cisco SiliconOne 芯片家族

博通 StrataXGS 芯片系列

博通（Broadcom）是以太网交换芯片产品的主要供应商，在大型数据中心

场景博通占据了绝大多数的市场份额。博通在交换机芯片领域推出了两大产品线：StrataDNX 和 StrataXGS，其中 DNX 系列主要用于大缓存及大表项的核心交换机场景，StrataXGS 则用于常规核心交换机场景及数据中心交换机场景。根据使用场景对端口速率和端口缓存及表项大小要求的不同，StrataXGS 主要包括 Trident 和 Tomahawk 两个系列，前者功能特性丰富，后者吞吐性能强大。2017 年博通对 Trident3-X7 交换芯片升级支持网络可编程，并在后续的 TD4 系列芯片均支持可编程特性。为了丰富和培育网络可编程生态，2019 年博通发布 TD4 交换芯片的同时对外发布了 NPL 网络编程语言，可基于“TD4 芯片+NPL 语言”进行网络可编程开发。

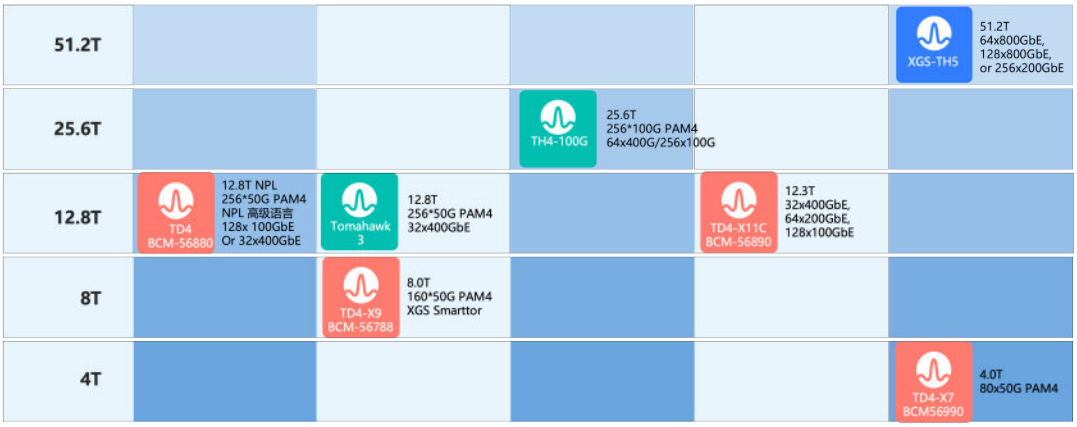

图 4-5 博通 StrataXGS 芯片家族

NPL 可编程支持的 Tile-mode 模式具备很好的可编程灵活性，只需要一套代码就可以涵盖多个不同的场景。基于 Tile-mode 特性可以实现不同应用场景下表项资源的动态灵活调整，比如在 L3 模式下可以实现大的路由表，在 L2 模式下可以实现大的 MAC 地址表等，一套 NPL 代码就可以同时实现多种场景特性，通过 SDK 的动态配置就可以实现应用模式的灵活调整切换。

随着可编程交换芯片应用的推广，博通在 2020 年针对应用最广泛的网关需求场景推出了 Trident4 SmartToR 芯片。SmartToR 芯片硬件表规格相对其他 Trident 芯片而言，提升了一个数量级，可实现数百万级的路由转发表和百万+级别的五元组流表，加上基于不同网络功能需求灵活重构转发表资源的能力（比如，灵活分配 MAC 表、主机路由表资源和流表等），Trident4 SmartToR 芯片适用于 NAT 网关、DDOS 攻击检测、网络分流，负载均衡、VXLAN 网关等应用场景，特别是有大规模路由转发表需求的无状态 SDN 网关的应用场景。

博通 StrataDNX 芯片系列

StrataDNX 系列芯片秉承一贯的可编程性，特别是从现已广泛部署的 Jericho2 一代开始，增强了其可编程性的开放程度。该芯片家族及后续产品提供了业界独一无二的可编程架构，除了管线节点可编程外，还可以进行管线延展，在增加了处理流程的同时而没有损失任何转发性能。另外，Jericho2 支持模块化表项结构，所有表项共享同一块物理缓存，极大增加了片上资源的使用效率。管线上所有处理节点可以并行访问，根据不同的应用场景进行逻辑表项的灵活划分，使得同样的硬件可以应用在完全不同的使用场景。

目前行业中量产的可编程交换芯片通常只支持固定数量的“Stage”，用户则在这些固定的“Stage”下进行流表的编程开发。但是，尽管在每个“Stage”下可以通过并行查找来增加支持的数据面协议数量，然而，当需要支持数据面协议数量较多，且复杂度较高的情况时，有限的“Stage”设计仍然无法完全满足需求，毕竟很多“Stage”的编程操作需要上一个 Stage 处理得到的结果。为了

解决这个问题，Jerico2 系列芯片基于 PEM (Programmable Element Matrix) 特性来延展 pipeline，增加新的 Stage，并且不会对带宽性能造成任何影响。而且可以执行算术运算（比如加法、减法、乘法），条件比较（比如==, !=, <, > 等等），逻辑以及比特操作等等。这种设计带来了两个好处：首先整个流水线的“Stage”数量可以通过引用可编程资源池被极大的扩展，解决了“Stage”数量固定的问题；除此之外，用户可以基于博通公司预置的成熟流水线设计之上，增加自己设计的功能，从而极大的降低功能风险并缩短开发时间。

在传统可编程交换芯片中，一个硬件存储资源块区（通常为 SRAM 和 TCAM）只能被分配到一个“Stage”中，所以用户需要对这些硬件资源进行非常仔细的管理和规划。为了解决这个问题，博通公司开发了 MDB (Modular Database) 模块化数据库架构，对 SRAM 和 TCAM 资源进行了抽象管理，允许一个网络应用的流表存储空间从另一个网络应用的存储空间中“借用”资源，来提高全局资源利用率。最为激动人心的是，这个全局的硬件存储资源池可以被所有的“Stage”引用，而不再需要考虑和设置资源和“Stage”之间的映射关系。博通公司提供了 MDB compiler 工具，通过友好的界面工具提供用户自行分配硬件资源的功能。

综上所述，目前基于网络可编程芯片技术实现无状态网关有三条技术路线可供选择，分别是：“Tofino+P4”、“SiliconOne+P4”、“TD/JR+NPL”。由于无状态网关对于大规模表项资源的需要，因此表项容量太小的可编程交换芯片无法满足大规模云网络的需求。

中国移动
China Mobile

缩略语列表

缩略语	英文全名	中文解释
SDN	Software Defined Network	软件定义网络
NFV	Network Function Virtualization	网络功能虚拟化
VPC	Virtual Private Cloud	虚拟私有云
IGW	Internet Gateway	互联网网关
VGW	VPC Gateway	VPC 网关
SLB	Server Load-Balancing	服务器负载均衡
NAT	Network Address Translation	网络地址翻译
DR	Direct Routing	直接路由
DSR	Direct Sever Return	直接服务器返回
LVS	Linux Virtual Server	Linux 虚拟服务器
DPDK	Data Plane Development Kit	数据平面开发套件
CPU	Central Processing Unit	中央处理器
DPU	Data Processing Unit	数据处理器
IPU	Infrastructure Processing Unit	基础设施处理器
FPGA	FieldField Programmable Gate Array	现场可编程逻辑门阵列

ASIC	Application Specific Integrated Circuit	专用集成电路
OVS	Open Virtual Switch	开放虚拟交换机
ACL	Access Control List	访问控制列表
SRAM	Static Random Access Memory	静态随机存取存储器
TCAM	Ternary Content Addressable Memory	三态内容寻址存储器
NP	Network Processor	网络处理器
DDR	Double Data Rate SDRAM	双倍数据速率 SDRAM
SDK	Software development kit	软件开发套件
API	Application Programming Interface	应用程序编程接口
IPDK	Infrastructure Programmer Development Kit	基础设施编程开发套件
DOCA	Data-Center-Infrastructure-On-A-Chip Architecture	芯片架构上的数据中心基础架构开发平台
NPL	Network Programming Language	网络编程语言
UPF	User Plane Function	用户面功能
GTP	GPRS Tunneling Protocol	GPRS 隧道协议
LPM	Longest Prefix Match	最长前缀匹配
MMU	Memory Management Unit	内存管理单元
ECMP	Equal-Cost Muti-pathing	等价多路径

DLB	Dynamic Load-balancing	动态负载均衡
MDB	Modular Database	模块化数据库
PEM	Programmable Element Matrix	可编程要素矩阵
VXLAN	Virtual eXtended LAN	虚拟扩展局域网
VRF	Virtual Routing and Forwarding	虚拟路由转发
VPN	Virtual Private Network	虚拟私有网


中国移动
China Mobile