



专题：新型网络技术

可编程网络数据平面技术进展

张昕怡¹, 潘恒², 谢高岗^{1,3}

(1. 中国科学院计算机网络信息中心, 北京 100190;
2. 中国科学院计算技术研究所, 北京 100190; 3. 中国科学院大学, 北京 100049)

摘要: 网络数据平面执行数据包处理转发, 是网络性能的决定性因素之一。大带宽、低时延、可持续演进的网络基础设施需要构建高效可编程的网络数据平面。首先, 介绍数据包处理转发模型, 并以此为基础概述网络数据平面在性能与可编程性面临的关键挑战。然后, 从数据包查找算法理论与软硬件协同实现机制出发, 详细论述其基本思路及关键核心技术进展以应对上述关键挑战。最后, 探讨高效可编程数据平面的未来发展趋势与技术演进路线。

关键词: 网络可编程数据平面; 软硬件协同设计; 数据包处理; 网络内计算

中图分类号: TP393

文献标志码: A

doi: 10.11959/j.issn.1000-0801.2022131

Progress in programmable network data plane

ZHANG Xinyi¹, PAN Heng², XIE Gaogang^{1,3}

1. Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China
2. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
3. University of Chinese Academy of Sciences, Beijing 100049, China

Abstract: Network data plane performs packet processing and forwarding, which is one of the core factors determining the network performance. High bandwidth, low latency and rapid evolution of the network infrastructure require building an efficient and programmable network data plane. The packet processing and forwarding model was firstly introduced, and on this basis, the challenges of the network data plane in the performance and programmability were outlined. Then the basic ideas and core technology progress were elaborated to deal with the above challenges by analyzing the theory of packet processing algorithms and hardware-software codesign implementations. Finally, the trend and technical route of constructing an efficient programmable data plane were discussed.

Key words: programmable network data plane, hardware-software codesign, packet processing, in-network computing

0 引言

互联网已成为人类社会的基础设施, 通过数

据包查找转发, 实现人机物互联。数据平面执行数据包查找转发, 是网络性能的决定性因素之一。4K/8K 超高清视频点播、增强现实/虚拟现实

收稿日期: 2022-04-30; 修回日期: 2022-06-07

基金项目: 国家重点研发计划项目 (No.2019YFB1802800)

Foundation Item: The National Key Research and Development Program of China (No.2019YFB1802800)

(augmented reality/virtual reality, AR/VR) 及物联网等应用^[1-3]的发展, 对网络提出大带宽、广连接和低时延等性能需求。受限于芯片器件工艺, 仅通过硬件升级和设备堆叠无法满足服务质量需求。为了解决上述问题, 软件定义网络 (software defined network, SDN)、网络切片、算网融合及新型网络体系结构被广泛研究与应用^[4-6]。上述技术要求数据平面在保障高效转发的同时, 提供查找规则、匹配运算、转发队列可编程能力, 因此, 网络数据平面查找算法与实现机制都面临新挑战。

1 数据包处理转发模型

数据平面的基本操作是数据包查找转发: 接收端口收到数据包后, 解析数据包相关字段, 匹配规则表; 根据精确匹配、最长前缀查找或正则表达式匹配结果, 执行数据包修改、转发或者丢弃等操作。不同的网络设备功能具有不同的规则表与匹配操作。例如, 交换机由媒体访问控制 (media access control, MAC) 表实现精确匹配, 路由器查找路由表执行最长前缀匹配, 防火墙查找五元组实现数据包分类, SDN 交换机查找多级流表实现转发, 5G 云核心网通过报文检测规则 (packet detection rule, PDR)、统一资源定位符 (uniform resource locator, URL) 规则匹配实现流量计费。

数据包查找转发是典型的 N 维空间点定位问题, 构造最优的查找决策树已被证明是多项式复杂程度的非确定性完全 (non-deterministic polynomial complete, NPC) 问题^[7]。IPv6 的广泛部署、规则表条目增加、网络切片与虚拟化转发表增加、SDN 流表字段增加与高频更新, 对数据平面性能与可编程性提出新需求, 而可编程性也进一步对性能提出新挑战。

1.1 数据平面性能

数据包查找模块转发速度将决定网络设备的

转发速率、队列长度与丢包率, 是网络性能的决定性因素之一。在 5G/B5G 网络环境中, 虽然无线侧空口传输的性能得到大幅度提升, 但核心网处理能力和效率受限于数据包处理转发的性能, 成为网络传输的性能瓶颈。面对当前网络环境中业务流量、系统规模以及功能复杂性的增长, 数据平面性能面临以下挑战。

- 查找速度方面, 大带宽网络需要数据包高速查找转发。当链路带宽为 100 Gbit/s 时, 按照最小以太网帧计算, 线速数据包转发吞吐量需达到约 150 Mbit/s, 即帧间时延约为 7 ns。实际上, 末级缓存 (last level cache, LLC) 的访问通常需要约 40 ns, 若 LLC 未命中, 访问内存则需要约 100 ns, 通用处理平台上的实现线速转发成为挑战。
- 存储容量方面, 目前骨干网路由器 IPv4 路由表项达到百万级, IPv6 路由表项达到十万级; 云计算平台虚拟路由表规模达到千万级; 云化 5G 核心网需为数十万用户的并发会话独立配置用户策略, 将在数据平面产生千万级的规则。大规模规则需要大存储空间, 增加设备制造成本与能耗, 也会因内存墙的存在诱发访存瓶颈, 使系统整体处理性能下降。
- 更新速度方面, 软件定义网络与网络功能虚拟化为网络带来极大的灵活性与可编程性, 同时, 网络状态和网络功能将频繁动态变化。在 5G 核心网和云数据中心, 业务的多样性加剧了网络的动态性和不确定性, 系统中规则更新的频率大幅提升。异步更新会产生大量的查找错误, 从而使重传等严重影响传输性能的问题; 同步更新则会直接阻塞查找进程, 降低整体处理性能。基于决策树数据包分类算法具有高更新时间复杂度, 要在保持高速查找的同时实现动态增量更新, 挑战巨大。



1.2 数据平面可编程性

互联网体系结构、协议与业务的持续创新需要可编程数据面提供执行、部署与验证环境。早期的可编程数据面主要基于通用处理器平台，数据包处理逻辑以软件方式部署执行，具有良好的灵活性与可编程性。近年来，虽然高性能输入/输出（input/output, I/O）架构（DPDK^[8]、Netmap^[9]、PF_RING ZX^[10]等）取得进展，但通用 CPU 的数据包处理性能有限。基于可编程交换芯片^[11]的网络数据面虽然可提供部分基础编程能力与 T 级处理能力，使用户通过特定编程语言构建复杂的网络应用，但仍面临以下挑战。

（1）统一抽象

基于可编程交换芯片的网络数据面通常采用“Match-Action”流水线的抽象结构，可实现无状态网络功能，但难以完整实现带状态的网络业务，导致应用范围受限。如何在现有数据面抽象基础之上进一步延伸扩展，构建统一、完整的抽象模型，支撑带复杂状态数据包运算，需要设计新的数据包处理抽象与范式。

（2）资源受限

可编程交换芯片的计算能力、存储资源、数据面流水线等多维资源受限，限制了数据面的数据包处理能力，无法支持复杂的数据包操作。如何在空间固定且受限的交换芯片上放置更多的计算逻辑、存储器件以及流水线结构，不仅是芯片制造工艺所面临的巨大挑战，更是交换芯片核心算法设计与实现的系统性难题。

（3）安全威胁

网络的开放性与可编程性会引入“人”的因素，将增加出错的风险，影响网络的可靠性与稳定性；同时，管控分离架构易造成拒绝服务攻击，导致转发性能下降。如何对数据面部署的网络策略进行实时异常检测与功能验证，对恶意网络流量进行安全隔离与处理，保障网络数据面安全是一个关键性挑战。

2 高效数据包查找算法

网络数据包查找转发是网关、交换机、路由器、防火墙等关键网络设备的核心功能，其性能直接决定了相关设备的处理能力、成本以及稳定性。随着网络带宽的增长、网络规模的扩大以及网络动态性的增加，现有方案难以满足大带宽、大容量、灵活动态的新需求。因此，创新查找转发关键算法和实现机制，突破现有技术性能瓶颈，是实现高效网络数据平面传输与计算的关键。

2.1 最长前缀查找算法

最长前缀查找是路由器的基本操作，随着骨干路由转发信息库（forwarding information base, FIB）的规模不断扩大和网络带宽的迅速增长，单域最长前缀查找算法需要不断提升查找速率并降低路由转发表存储开销。云计算虚拟路由器平台由于多个虚拟路由器共存，存在多个 FIB，需要在保证低存储开销的同时，提升查找与更新速度。随着 IPv6 技术的广泛部署，IP 地址前缀长度从 IPv4 中的 32 bit 变为 128 bit，查找时间与存储开销进一步增加。提升路由查找性能，可以基于算法理论构建低存储开销与查找时间复杂度的算法，还可以利用规则的分布和特征优化查找结构。对于 IPv4 最长前缀查找，可以利用 IP 地址前缀规则非均匀分布特征和片上/片外存储的性能特点，通过对查找流程和规则存储拆分，实现存储空间高效利用的同时达到高速的查找速率^[12]；对于多虚拟路由表，可以通过挖掘规则前缀特征，设计高效的数据结构和合并方案，实现面向虚拟路由器的高性能、支持快速更新、可扩展的 IP 地址查找方法^[13]；对于 IPv6 最长前缀查找，同样可以基于规则集的分布，优化查找数据结构降低访存次数，提升查找速率^[14]。

2.2 数据包分类算法

数据包分类算法基于多域规则集（如五元组）对数据包进行查找匹配，是防火墙、入侵检测、

负载均衡系统、虚拟专用网络（virtual private network, VPN）等网络设备的基本操作。随着网络规模的扩大和网络功能的复杂化，规则的急剧增加使数据包分类算法面临的存储压力和性能挑战更加严峻。通过预先设定的方案划分整个规则空间并构造决策树，能够实现查找结构体的快速构建^[15-16]，但是由于忽略了规则的特征，构建决策树的过程可能产生规则的大规模复制进而增加存储开销；此外，不同的决策树算法在不同的规则集会出现较大的性能波动和差异。通过设计估算不同算法存储开销的评估预测模型^[17]，基于规则特征动态选择最优的方案构造决策树，能够显著减少规则复制，降低存储开销，同时提升查找性能。

2.3 流表查找算法

流表查找算法本质属于多域规则匹配，只因软件定义网络与网络功能虚拟化场景规则更新频繁，需要流表查找算法能够支持规则集的快速增量更新。传统的流表查找算法基于哈希表^[18]实现规则的高速更新，但是数据包查询时需要遍历哈希表，因此查询速率缓慢。虽然基于预先设定的策略对哈希表进行合并^[19]可以减少哈希表的数目，但由于忽略了规则的特征，可能会产生较多的哈希冲突数从而增加验证时间，影响查询速率。因此，如何在有效地减少哈希表的数目同时降低哈希冲突数，是实现高效流表查找算法的关键。要解决上述问题，一方面可以将合并哈希表产生的冲突数转化为合并概率（即冲突数越大合并的概率越小），进而指导哈希表的合并^[20]；另一方面，可以利用人工智能中监督学习的方法，通过线下筛选合并方案并完成训练模型，指导线上基于哈希表的查找结构体的构建，在减少哈希表数目的同时，有效控制哈希冲突概率，实现规则的高速增量更新^[21]。

虽然软件算法的创新始终是实现高效数据平面的技术路线之一，但算法的执行效率与硬件体

系结构密切相关。基于硬件体系结构设计算法，能够高效充分利用硬件资源从而提升数据平面性能。因此，通过软/硬协同的方式实现高效可编程的网络数据平面成为当前研究的热点。

3 软/硬件协同实现机制

专用硬件平台受限于芯片处理逻辑的设计，不同平台的可编程性通常与性能成反比，硬件实现的性能与可编程性如图 1 所示。在 CPU 通用软件平台上，CPU 高度灵活，能够实现复杂计算场景的需求，但是 CPU 没有针对数据包处理设计实现专用的加速逻辑，性能较硬件转发设备差。此外，云平台中 CPU 及内存等资源由多业务共享竞争，无法保障数据平面中负责数据包处理转发程序的性能。现场可编程逻辑门阵列（field programmable gate array, FPGA）芯片内集成大量的基本门电路以及存储器，用户通过更新 FPGA 配置文件定义门电路以及存储器之间的连线，被广泛用作硬件可编程加速平台。网络处理器（network processor, NP）通常由若干微码处理器和若干硬件协处理器组成，可以基于微码编程，同时能够实现高速数据包的处理。专用集成电路（application specific integrated circuit, ASIC）作为定制加速平台，其硬件电路不可改变，因此也不具备可编程的能力。但是 ASIC 面向特定应用场景定制开发，因此仅能够在对应的应用场景发挥硬件平台强劲的性能。

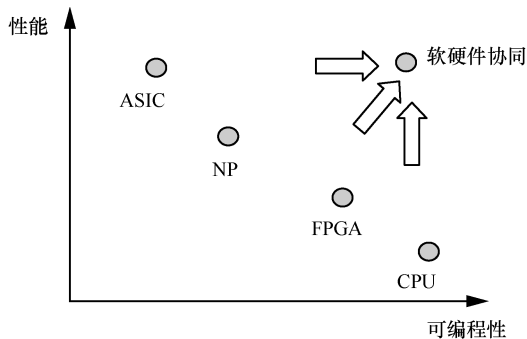


图 1 硬件实现的性能与可编程性



通过上述分析可知,要实现高性能且可编程的数据平面,需要将通用平台的灵活性和可编程性与专用硬件的高效性深度融合,软/硬件协同加速。

3.1 数据平面可编程

基于不同的数据平面架构,通过开发高级编程语言能够降低网络数据平面编程门槛,为网络数据平面的研究和发展带来更多契机。Click^[22]基于C++实现数据包在CPU中的处理逻辑,PX^[23]则是面向FPGA平台的数据包处理语言,而可编程协议无关报文处理(programming protocol-independent packet processors, P4)^[24]作为一种协议无关的高级编程语言,能够屏蔽可编程交换芯片底层硬件逻辑,基于芯片提供的接口对网络数据平面进行编程,目前应用最为广泛。P4通过基于高层抽象语言对数据包处理流程进行自定义,实现对数据包处理流程逻辑即时编程的可重配置性,基于自定义的数据处理逻辑实现协议无关性,并且在代码编写过程中,P4无须关心底层设备实现细节,因而具备目标设备无关性。借助P4带来的数据平面编程能力,网络研究人员可以实现路由器、防火墙等网络设备功能与网络协议,加速网络创新。

同时,基于可编程芯片和P4,网络具备初步的计算能力,这使传统以CPU/GPU为中心的计算模式逐渐转变为以数据为中心的计算模式,网络与计算进一步深度融合,网络内计算(in-network computing)这一新型技术得以兴起。网络内计算的核心思想在于通过网络侧与端侧的协同,使网络数据在传输的过程中边转发边计算,从而降低端侧计算压力,提升网络传输的吞吐量。当前,网络内计算技术的发展主要集中在两个方面。其一是对网络算力的增强,从而使网络具备处理更为复杂业务与数据的能力。例如,NetFC^[25]设计了基于多表查询的在网浮点数计算方法;P4AES^[26]使网络具有数据加解密能力;而ZipLine^[27]实现了网络数据的在网压缩/解压缩。其二是通过网络内

计算技术加速传统网络功能或典型分布式应用。例如,Beamer^[28]将负载均衡部分功能实现在可编程交换机上,从而能够处理40 Gbit/s的HTTP流量;SwRL^[29]在可编程交换机上实现了高性能可扩展流量限速器;NetCache^[30]利用可编程交换机对网络的key-value存储应用进行缓存及快速响应,其性能可以达到每秒20亿次查询;SwitchML^[31]和ATP^[32]提出了基于可编程交换机的网络内梯度聚合机制,加速分布式训练效率;NetSHA^[33]将分布式搜索中的答案约减卸载到可编程交换机完成,极大提升检索效率。

3.2 高性能网络处理

传统的数据包处理方式基于CPU中断,即网卡驱动接收数据包后通过中断通知CPU处理,然后由CPU拷贝数据并交给协议栈。随着网络带宽的增加,海量数据包的处理将产生大量CPU中断,使系统时延上升、吞吐量下降,同时降低基础架构的整体效率。DPDK^[8]、Netmap^[9]、PF_RING ZX^[10]等架构基于内核旁路(kernel bypass)技术实现高性能I/O。其中,DPDK可提供高性能的数据包处理库和用户空间驱动程序,是目前主流的高性能I/O架构。不同于传统数据包处理使用CPU中断方式,DPDK采用轮询方式实现数据包处理过程。DPDK重载了网卡驱动,该驱动在收到数据包后不产生中断通知CPU,而是将数据包通过零拷贝技术存入内存,应用层程序通过DPDK提供的接口直接从内存读取数据包。这种处理方式节省了CPU中断时间、内存拷贝时间,并向应用层提供了简单易行且高效的数据包处理方式。

(1) CPU指令优化算法

通用平台上的网络数据平面的查找算法由软件实现。算法或者数据结构的优化可以提升算法查找性能,进而优化网络数据平面的处理能力。而算法的高效性与体系结构密切相关,利用体系结构特性或CPU指令进一步优化软件算法。针对数据平面中的单域最长前缀查找算法,可基于

CPU 体系结构和缓存调度特点, 利用访存局部性实现高通量算法。对基于决策树的数据包分类算法, 可利用 CPU 比特位操控指令 PEXT^[34]对每个树节点制定有效的切割规则, 构建高效的查找数据结构并提升查找速率。

(2) 网卡卸载

智能网卡 (smart network interface card, Smart NIC) ^[35-38]通过传统以太网接口和硬件总线之间, 嵌入具有计算和存储功能的网络处理器或现场可编程逻辑门阵列, 使其具有高度可编程化定制的特点, 提升计算和存储能力。通过使用智能网卡协助 CPU 处理网络负载 (包括 TCP 卸载、VxLAN/GRE 隧道报文卸载、IPSec^[39]加/解密的卸载、防火墙卸载等), 能够有效减轻宿主机 CPU 负载。以 IPSec 在智能网卡上的卸载为例, IPSec 是一组基于网络层的安全通信协议族。通过对 IP 的分组进行加密和认证来保护 IP 的网络传输协议簇, 已部署于 VPN、无线网络和安全网络虚拟化隧道。当开启 IPSec 功能后, IPSec 处理模块会在收到的数据包 IP 头和高层协议头之间插入经复杂计算得到的 IPSec 头。而随着网络带宽的增加, 频繁的加/解密操作会消耗大量 CPU 资源, 严重影响系统性能。通过将 IPSec 中的加/解密任务卸载到智能网卡, 虽然部分操作仍需要在软件侧完成, 但是利用智能网卡硬件优势能够显著提升系统性能。

面向数据中心的专用处理器 (data processing unit, DPU) 最早由硅谷创业公司 Fungible^[40]提出, 英伟达公司 (NVIDIA) 在收购迈勒罗斯科技公司 (Mellanox Technologies) 后, 推出 BlueField 系列 DPU^[41]。BlueField DPU 在传统智能网卡上嵌入 CPU, 通过卸载控制平面的任务和一些灵活复杂的数据平面任务, 达到释放宿主 CPU 资源、提升网络性能的目的。以 DPU 加速虚拟网络交换机 (open vSwitch, OVS) ^[42]为例, 不同于 Mellanox ConnectX-5^[43]在智能网卡实现相对简单的可编程数据路径, 控制平面仍需要在宿主 CPU

处理。即当收到报文后, 检查是否在快速路径已缓存该类报文的处理规则, 命中则直接按照对应的规则执行操作, 否则传送至软件虚拟交换机。

BlueField DPU 中 OVS^[44]直接部署在其上 ARM (advanced RISC machine) 架构的 CPU 中, 而 OVS 中的快速路径可卸载到网络硬件加速器 eSwitch, 该卸载方案可降低数据面和控制面之间通信的时延, 加速网络数据平面的处理。

4 发展趋势

4.1 数据包匹配算法

对于高效数据平面算法的实现, 可以分别从优化算法实现与算法理论创新两个方面开展。

在优化算法实现方面, 由于规则集分布特性和流量特征影响算法性能, 通过深入挖掘规则分布特性与流量特征优化算法, 能够极大提升算法性能。随着人工智能技术的发展与进步, 基于人工智能的技术实现流量和规则特征提取及算法优化, 是实现高效算法的关键解决思路和研究方向。对于高效算法的实现, 除了分别从时间复杂度和空间复杂度优化代码的执行效率和空间存储, 由于算法的执行效率与硬件体系结构密切相关, 还可以进一步从体系结构、CPU 指令执行的角度实现算法与系统的深度融合。

在算法理论创新方面, 通过针对具体的应用场景设计新型数据结构, 能够进一步提升网络性能。Sketch^[45-46]通常用于数据平面测量, 能够在高速网络环境中实时地存储流量特征信息, 并且仅占用较小的存储空间, 具备在理论上可证明的估计精度与内存的平衡特性。Learned index^[47-48]通常用于索引查询, 通过利用机器学习技术得到更好的范围索引, 使用机器学习模型代替传统的 B 树索引, 从而在实现存储高效压缩的同时保证较高的查询速率。

优化创新数据包匹配算法是大幅提升网络数据平面性能的关键。如何在使网络更加智能化的



基础上,更高效准确地提取流量和规则特征,基于具体应用场景对关键数据结构进行创新,将是网络数据平面算法的设计思路和研究方向。

4.2 计算内网络

随着云数据中心、虚拟化等技术的日趋成熟,虚拟服务器逐渐替代裸金属服务器为不同客户提供计算、存储等差异化服务,达到高效利用不同类型资源的目的。单物理机或多物理机中的不同虚拟服务器之间需要通过虚拟网络进行互联互通、协同计算。虚拟网络的性能已逐渐成为影响协同计算效率的关键因素。

高性能虚拟化网络的构建需要从算法与系统两个层面分别考量与设计。在算法层面上,需要设计灵活高效可扩展的查找转发机制与数据结构,可针对不同网络业务类型,快速确定网络数据包的虚拟转发路径与转发端口,提升转发效率;在系统方面,需要设计高效的虚拟 I/O 与缓存机制,减少网络数据包在虚拟转发过程中的数据拷贝,提升整体的缓存命中率,提高系统性能。但无论是算法还是系统,需要充分挖掘新型网络硬件的性能潜力,通过软/硬协同设计,提升计算设备中的虚拟网络性能。虚拟转发、高效可靠虚拟 I/O 以及软/硬协同系统设计架构与关键算法是未来需要解决的问题。

4.3 网络内计算

随着可编程网络设备的不断演进与革新,特别是 Mellanox 智能网卡、Intel Tofino 可编程交换机的出现,数据包在网络传输过程中被计算、处理和转发。受限于物理极限,网络内单点设备的计算能力、片上存储、流水线 stage 等多维资源受限,无法满足复杂业务的网络内计算需求,网络内多点协同计算成为算网融合发展的一个重要趋势。

网络内协同计算的核心在于自动确定数据包的计算内容、逻辑与位置,在满足灵活性的基础上保障高性能。为此,构建一个面向服务定制的数据包计算框架显得尤为重要。该计算框架需要

对差异化业务进行全等级多粒度的分析与抽象,形成数据包计算基本操作单元能力集基于开放的编程接口及模板,不同网络业务可通过数据包计算基本操作单元进行灵活、快速构建,利用框架的运行时系统进行编译、解析,并将数据包计算基本操作单元精准映射到底层异构平台进行高效执行。数据包计算统一编程模型与编程语言、数据包计算实时编译系统以及异构数据面高效实现机制等是亟待解决的问题。

5 结束语

高效可编程数据平面是重要的新型组网实现技术。本文介绍了可编程数据面的转发模型及面临的关键性挑战,梳理了高效数据包查找算法与软/硬件协同实现机制等不同解决途径的技术路线及发展趋势。随着可编程网络技术的不断发展,网络数据面能力得到极大提升,将从传统“传输管道”逐渐向以“数据包计算”为核心的传算一体执行平台演进。因此,构建统一的数据包计算模型、框架与实现机制,满足数据包处理转发按照业务负载在网络与计算内自适应执行是未来可编程数据面发展的重要方向之一。

参考文献:

- [1] AGIWAL M, ROY A, SAXENA N. Next generation 5G wireless networks: a comprehensive survey[J]. IEEE Communications Surveys & Tutorials, 2016, 18(3): 1617-1655.
- [2] NIGHTINGALE J, SALVA-GARCIA P, CALERO J M A, et al. 5G-QoE: QoE modelling for ultra-HD video streaming in 5G networks[J]. IEEE Transactions on Broadcasting, 2018, 64(2): 621-634.
- [3] SIRIWARDHANA Y, PORAMBAGE P, LIYANAGE M, et al. A survey on mobile augmented reality with 5G mobile edge computing: architectures, applications, and technical aspects[J]. IEEE Communications Surveys & Tutorials, 2021, 23(2): 1160-1192.
- [4] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, et al. OpenFlow[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74.
- [5] HONG C Y, MANDAL S, AL-FARES M, et al. B4 and after: managing hierarchy, partitioning, and asymmetry for availability

- ty and scale in Google's software-defined WAN[C]// Proceedings of SIGCOMM '18: Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. [S.l.:s.n.], 2018: 74-87.
- [6] HAN B, GOPALAKRISHNAN V, JI L S, et al. Network function virtualization: challenges and opportunities for innovations[J]. IEEE Communications Magazine, 2015, 53(2): 90-97.
- [7] HYAFIL L, RIVEST R L. Constructing optimal binary decision trees is NP-complete[J]. Information Processing Letters, 1976, 5(1): 15-17.
- [8] DPDK. Developer quick start guide[EB]. 2022.
- [9] RIZZO L. Netmap: a novel framework for fast packet I/O[C]// Proceedings of 21st USENIX Security Symposium. [S.l.:s.n.]. 2012:101-112.
- [10] Ntop. PF_RING: high-speed packet capture, filtering and analysis[EB]. 2022.
- [11] Intel. 探索英特尔® 智能结构处理器的强大功能[EB]. 2022. INTEL. Discover the power of the Intel® intelligent fabric processor[EB]. 2022.
- [12] YANG T, XIE G G, LI Y B, et al. Guarantee IP lookup performance with FIB explosion[C]//SIGCOMM '14: Proceedings of the 2014 ACM Conference on SIGCOMM. New York: ACM Press, 2014: 39-50.
- [13] LUO L Y, XIE G G, SALAMATIAN K, et al. A trie merging approach with incremental updates for virtual routers[J]. 2013 Proceedings IEEE INFOCOM, 2013: 1222-1230.
- [14] ZHANG C Y, XIE G G. Using XorOffsetTrie for high-performance IPv6 lookup in the backbone network[J]. Computer Communications, 2022(181): 438-445.
- [15] VAMANAN B, VOSKUILEN G, VIJAYKUMAR T. EffiCuts: optimizing packet classification for memory and throughput[J]. ACM SIGCOMM Computer Communication Review, 2010, 40(4): 207-218.
- [16] YINGCHAREONTHAWORNCHAI S, DALY J, LIU A X, et al. A sorted-partitioning approach to fast and scalable dynamic packet classification[J]. IEEE/ACM Transactions on Networking, 2018, 26(4): 1907-1920.
- [17] HE P, XIE G G, SALAMATIAN K, et al. Meta-algorithms for software-based packet classification[C]//Proceedings of 2014 IEEE 22nd International Conference on Network Protocols. Piscataway: IEEE Press, 2014: 308-319.
- [18] SRINIVASAN V, SURI S, VARGHESE G. Packet classification using tuple space search[C]//Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication - SIGCOMM '99. New York: ACM Press, 1999.
- [19] DALY J, BRUSCHI V, LINGUAGLOSSA L, et al. TupleMerge: fast software packet processing for online packet classification[J]. IEEE/ACM Transactions on Networking, 2019, 27(4): 1417-1431.
- [20] ZHANG X Y, SALAMATIAN K, XIE G G. Improving open virtual switch performance through tuple merge relaxation in software defined networks[J]. IEEE Transactions on Network and Service Management, 5592(99): 1.
- [21] ZHANG X Y, XIE G G, WANG X, et al. Fast online packet classification with convolutional neural network[J]. IEEE/ACM Transactions on Networking, 2021, 29(6): 2765-2778.
- [22] KOHLER E, MORRIS R, CHEN B J, et al. The click modular router[J]. ACM Transactions on Computer Systems, 2000, 18(3): 263-297.
- [23] BREBNER G, JIANG W R. High-speed packet processing using reconfigurable computing[J]. IEEE Micro, 2014, 34(1): 8-18.
- [24] BOSSHART P, DALY D, GIBB G, et al. p4[J]. ACM SIGCOMM Computer Communication Review, 2014, 44(3): 87-95.
- [25] CUI P L, PAN H, LI Z Y, et al. NetFC: enabling accurate floating-point arithmetic on programmable switches[C]//Proceedings of 2021 IEEE 29th International Conference on Network Protocols. Piscataway: IEEE Press, 2021: 1-11.
- [26] CHEN X Q. Implementing AES encryption on programmable switches via scrambled lookup tables[C]//SPIN '20: Proceedings of the Workshop on Secure Programmable Network Infrastructure. [S.l.:s.n.], 2020: 8-14.
- [27] VAUCHER S, YAZDANI N, FELBER P, et al. ZipLine: in-network compression at line speed[C]//Proceedings of the 16th International Conference on emerging Networking Experiments and Technologies. [S.l.:s.n.], 2020: 399-405.
- [28] OLTEANU V, AGACHE A, VOINESCU A, et al. Stateless datacenter load-balancing with beamer[C]//Proceedings of 15th USENIX Symposium on Networked Systems Design and Implementation. [S.l.:s.n.], 2018: 125-139.
- [29] HE Y C, WU W F, WEN X M, et al. Scalable on-switch rate limiters for the cloud[C]//Proceedings of IEEE INFOCOM 2021 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2021: 1-10.
- [30] JIN X, LI X Z, ZHANG H Y, et al. NetCache: balancing key-value stores with fast in-network caching[C]//Proceedings of the 26th Symposium on Operating Systems Principles. New York: ACM Press, 2017.
- [31] SAPIO A, CANINI M, HO C Y, et al. Scaling distributed machine learning with in-network aggregation[EB]. 2019.
- [32] LAO C, LE Y F, MAHAJAN K, et al. ATP: In-network Aggregation for Multi-tenant Learning[C]//Proceedings of 18th USENIX Symposium on Networked Systems Design and Implementation. [S.l.:s.n.], 2021: 741-761.
- [33] ZHANG P H, PAN H, LI Z Y, et al. Accelerating LSH-based distributed search with in-network computation[C]//Proceedings of IEEE INFOCOM 2021 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2021: 1-10.
- [34] INTEL. Development reference guides: _pext_u32/64[EB]. 2022.



- [35] MA X X, YAN, F, WANG Z, et al. Survey on smart network interface card[J]. Journal of Computer Research and Development. 2022, 59(1): 1-21.
- [36] INTEL. 700 series network adapters[EB]. 2022.
- [37] NETRONOME. Agilio open vSwitch TC user guide[EB]. 2022.
- [38] NAPATECH. Napatech smart NIC solution for hardware offload[EB]. 2022.
- [39] FRANKEL S, KRISHNAN S. IP security and internet key exchange document roadmap RFC 6071[EB]. 2011.
- [40] FUNGIBLE. The Fungible data processing unit[EB]. 2022.
- [41] NVIDIA. NVIDIA bluefield data processing units[EB]. 2022.
- [42] PFAFF B, PETTIT J, KOPONEN T, et al. The design and implementation of open vswitch[C]//Proceedings of 12th USENIX Symposium on Networked Systems Design and Implementation. [S.l.:s.n.], 2015: 117-130.
- [43] NVIDIA. OVS offload using ASAP² direct[EB]. 2022.
- [44] NVIDIA. Virtual switch on bluefield smart NIC[EB]. 2022.
- [45] CORMODE G, MUTHUKRISHNAN S. An improved data stream summary: the count-Min sketch and its applications[J]. Journal of Algorithms, 2005, 55(1): 58-75.
- [46] NAMKUNG H, LIU Z X, KIM D, et al. SketchLib: enabling efficient sketch-based monitoring on programmable switches[C]//Proceedings of 19th USENIX Symposium on Networked Systems Design and Implementation. [S.l.:s.n.], 2022: 743-759.
- [47] KRASKA T, BEUTEL A, CHI E H, et al. The case for learned index structures[C]//Proceedings of the 2018 International Conference on Management of Data. New York: ACM Press, 2018.
- [48] MARCUS R, ZHANG E, KRASKA T. CDFShop: exploring and optimizing learned index structures[C]//Proceedings of

SIGMOD '20: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2020: 2789-2792.

[作者简介]



张昕怡（1992- ），女，博士，中国科学院计算机网络信息中心在站博士后，主要研究方向为计算机网络系统结构、高性能网络数据平面。



潘恒（1990- ），男，博士，中国科学院计算技术研究所副研究员，主要研究方向为可编程网络与系统。



谢高岗（1974- ），男，博士，中国科学院计算机网络信息中心研究员，中国科学院大学教授、博士生导师，主要研究方向为计算机网络体系结构与系统。