

下一代网络处理器及应用综述*

赵玉宇^{1,2}, 程光^{1,2}, 刘旭辉^{1,2}, 袁帅^{1,2}, 唐路³



¹(东南大学 网络空间安全学院, 江苏 南京 211189)

²(教育部计算机网络和信息集成重点实验室(东南大学), 江苏 南京 211189)

³(湖南华芯通网络科技有限公司, 湖南 长沙 410000)

通讯作者: 程光, E-mail: gcheng@njnet.edu.cn

摘要: 网络处理器作为能够完成路由查找、高速分组处理以及 QoS 保障等主流业务的网络设备核心计算芯片, 可以结合自身可编程性完成多样化分组处理需求, 适配不同网络应用场景。面向超高带宽及智能化终端带来的网络环境转变, 高性能可演进的下一代网络处理器设计是网络通信领域的热点问题, 受到学者们的广泛关注。融合不同芯片架构优势、高速服务特定业务, 使得下一代网络处理器具备分组处理性能加速、动态配置硬件资源和服务应用智能化的特点。对利用新型可编程技术、面向新型网络体系结构以及针对新型高性能业务的下一代网络处理器设计方案与现有研究进行分析比较, 并对下一代网络处理器的工业化进程做了介绍; 提出了高性能可演进的下一代网络处理器体系架构, 通过软硬件协同分组处理流水线、多级缓存与分组调度、资源管理及编程接口等方面给出了架构设计细节, 研制了原型系统并对其性能进行了测试。明确了自主可控的网络处理器体系架构的发展方向和智能化应用场景, 讨论了未来可能的研究方向。

关键词: 网络处理器; 下一代; 高性能可演进; 自主可控

中图法分类号: TP393

中文引用格式: 赵玉宇, 程光, 刘旭辉, 袁帅, 唐路. 下一代网络处理器及应用综述. 软件学报, 2021, 32(2): 445-474. <http://www.jos.org.cn/1000-9825/6124.htm>

英文引用格式: Zhao YY, Cheng G, Liu XH, Yuan S, Tang L. Survey and applications of next generation network processor. Ruan Jian Xue Bao/Journal of Software, 2021, 32(2): 445-474 (in Chinese). <http://www.jos.org.cn/1000-9825/6124.htm>

Survey and Applications of Next Generation Network Processor

ZHAO Yu-Yu^{1,2}, CHENG Guang^{1,2}, LIU Xu-Hui^{1,2}, YUAN Shuai^{1,2}, TANG Lu³

¹(School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China)

²(Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing 211189, China)

³(Hunan Huaxintong Network, Changsha 410000, China)

Abstract: As the core computing chip of network equipment, network processor can complete the essential services such as routing lookup, high-speed packet processing, and QoS guarantee. Facing the transformation of network environment brought by ultra-high bandwidth and intelligent terminal, the design of the next generation network processor (NGNP) with high performance and evolution is a hot issue in the field of network communication, which is widely concerned by scholars. Combining the advantages of different chip

* 基金项目: 国家重点研发计划(2018YFB1800602); 赛尔网络下一代互联网技术创新项目(NGIICS20190101, NGII20170406); 教育部-中国移动科研基金(MCM20180506)

Foundation item: National Key Research and Development Program of China (2018YFB1800602); CERNET Innovation Program for Next Generation of Internet (NGIICS20190101, NGII20170406); Ministry of Education-China Mobile Research Fund (MCM2018 0506)

收稿时间: 2020-02-28; 修改时间: 2020-05-27, 2020-07-17; 采用时间: 2020-08-07; jos 在线出版时间: 2020-09-10

architectures and high-speed services, NGNP has the characteristics of accelerating packet processing, dynamic configuration of hardware resources, and intelligent service application. In this study, the existing research is analyzed and compared from the design of NGNP which using new programmable technology, new network architecture oriented and for new high-performance service. The industrialization process of NGNP is summarized. Finally, the high performance evolvable network processor (HPENP) architecture is proposed. By introducing the hardware and software collaborative packet processing pipeline, multi-level cache and packet scheduling, resource management and programming interface, the details of HPENP design are given and a prototype system is developed and its performance is tested. In this study, the development direction and intelligent application scenario of autonomously controlled network processor architecture are confirmed, and the possible research direction in the future is discussed.

Key words: network processor; next generation; high performance evolvable; autonomous and controllable

自 21 世纪以来,人类已经不再满足于只运用互联网进行简单的信息交互.随着终端设备的智能化,互联网承载的任务变得复杂多样.这种发展一方面提高了人类社会生产生活的效率,为人们带来了很大的便利以及好处,但另一方面,也加重了互联网中以路由器、交换机为主的各项基础设施的压力.尤其是在约束了特定服务质量(QoS)的情况下,网络设备处理速度随着 CPU 利用率增高而下降,使得网络传输效率降低,用户体验较差.当代互联网的发展,要求网络设备拥有强大的计算和存储能力,以便满足复杂网络环境以及特定任务.随着互联网用户需求的不断革新,人们对网络设备的可编程能力也提出了更高的要求.

网络处理器(network processor,简称 NP)就是为网络设备提供计算能力的核心器件.它是一种能够完成路由查找、协议分类、报文处理以及防火墙和 QoS 等各种任务的通信网络芯片^[1].当前的部分网络处理器具备了一定的可编程性,能够灵活地应对管理人员希望使能的功能需求.在计算能力方面,拥有多个微码处理器和硬件协处理器的 NP 能够对内存操作、路由表查找算法、拥塞控制以及主动队列管理进行一些标准操作.在下一代网络体系结构的提出与应用以及终端计算能力变强等硬件设备革新的技术潮流下,现有 NP 由于设计缺陷和人为疏漏,未能很好地兼顾灵活性和高性能,使得网络设备的能力始终滞后,无法满足未来网络通信设备应用场景的迫切需求.所以,网络处理器的研究设计始终是网络与通信方面的热点问题,其研究层次逐渐提高.由于中兴通讯股份有限公司(下文简称为中兴)、华为技术有限公司(下文简称为华为)等中国先进通信设备制造商接连遭受美国制裁或者禁运,无法获得 NP 芯片技术,网络处理器的国产化也成为国内业界学术界面临的紧急任务.

为了能够自适应地满足高性能和灵活应用场景的各项要求,使 NP 拥有多样化分组处理以及超高带宽的体系结构^[2],下一代网络处理器(next generation network processor,简称 NGNP)成为 NP 研究与设计的热点.NGNP 的设计核心思想是:在灵活性上拥有良好的可编程性,尽可能地满足高级语言编程;针对不断涌现的新业务,能够减少部署时间,提高效率;在性能上,能够利用加速引擎或者体系结构优势优化处理流程、处理时间;同时,在能耗、芯片尺寸上尽可能地轻量化,降低成本的同时,提高网络设备的综合能力(见表 1).

Table 1 Comparison between next generation network processor and traditional network processor

表 1 下一代网络处理器与传统网络处理器对比

	传统网络处理器	下一代网络处理器
可编程性	可简单编程,支持部分自定义功能	提供开放 API,支持高级语言编程
部署升级	在不改变硬件可用性情况下升级	按需定制,快速反应
转发性能	2.5G/s~5G/s	2Tb/s 以上
功耗情况	1W 左右	低于 0.1W
芯片工艺	0.18 μ m 工艺	小于等于 28nm

下一代网络处理器是一种借鉴新型可编程技术、面向新型网络体系结构或新型应用的设计理念和目标,并不是指某一种具体的芯片设计.当前,在 NGNP 设计目的指引下,国际国内的很多研究者进行了大量的研究工作.下一代网络处理器的各项内在设计关键技术也随着新型网络体系结构、可编程技术以及在网络环境复杂多变的高性能业务引导下进行革新和发展.学术界主要是提供 NGNP 的体系结构以及关键技术研究,产生了很多新的 NP 体系结构,也在协处理器设计上进行了突破,使其能够满足某一特定新型业务如优化数据转发、新型拥塞控制方法等.工业界则根据原有 NP 迭代更新产品,定向提高了芯片内 CPU 计算能力,或者加大带宽以满足当前应用需求.自 2010 年以来,有部分学者进行了 NGNP 技术的相关分析和研究,但审视现在已经公开发表的包括

科技论文和专利等资料,其中大部分只是介绍和阐述 NP 的特定新应用、新架构.国内外对下一代网络体系结构的分析和研究不够全面和深入,并且基本没有对 NGNP 的综述性的学术著作出现.2010 年,Albrecht 和 Carsten 等人^[3]发表专著,介绍了一种利用动态可重构的协处理 DynaCORE,以增加 NP 相关负载功能,但是该书没有宏观地提炼 NGNP 的相关体系结构.2016 年,Gadre、Geetanjaliy 以及 Matthews 等人^[4,5]系统性地提出了利用精简指令集计算机(reduced instruction set computing,简称 RISC)处理器作为高速网络接口的 NP 结构,但是该文献注重技术细节的阐述,针对新型网络体系结构下的 NP 设计分析简单,主要还是面向 NP 灵活性进行体系结构的技术更新,也缺少其他同类 NP 体系结构设计分析对比.

下一代网络处理器包括设备功能模块、体系结构、外围支撑以及协议与平台等内在关键技术,都是围绕新型网络架构技术及应用进行研究与设计.因此,本综述尝试对下一代网络处理器的关键技术设计研究进行全面的总结与归纳,按照利用新型可编程技术、面向新型网络体系结构、针对新型高性能业务的 3 类下一代网络处理器设计方法进行阐述,提出了下一代网络处理器发展的挑战,设计了一种下一代高性能可演进的网络处理器体系架构,实现了原型系统并对其进行了测试.

本文第 1 节介绍网络处理器的基本概念与面临的挑战.第 2 节~第 4 节按照 3 类方法分类阐述下一代网络处理器设计的主要指导思想.第 6 节提出下一代网络处理器的发展方向,并介绍一种高性能可演进的下一代网络处理器体系架构与实现方法.第 7 节对全文的工作进行总结.

1 网络处理器基本概念

1.1 网络处理器基本架构

网络处理器由多个处理器内核构成^[6],一般地,这些内核分为处理单元(processing element,简称 PE)、协处理单元(co-processor,简称 CoP)以及硬件逻辑管控单元(hardware logic block,简称 HLB).以上单元的配置方式、指令集设计、共享资源访问策略以及调度方法一般有两种处理机制,处理机制的不同,导致网络处理器的硬件逻辑结构不同:一种是流水线方式,如图 1 所示的 Yang 等学者^[7]提出的一种经典网络分组处理操作流程,各单元利用处理引擎无关的特点,通过流水线实现系统指令并行处理,完成分组相关操作;另一种是并行处理方式,由于使用并行处理要频繁地对分组状态进行更新,利用存储表格维护其状态,所需要的计算资源相对较多,采用并行处理的时机非常重要,使得每个处理单元完成相似的任务.并行处理架构使用 HLB 和 PE 集群关联,利用 PE 间并行、PE 内部并行或者 PE 和 CoP 并行方式提高处理器效率.

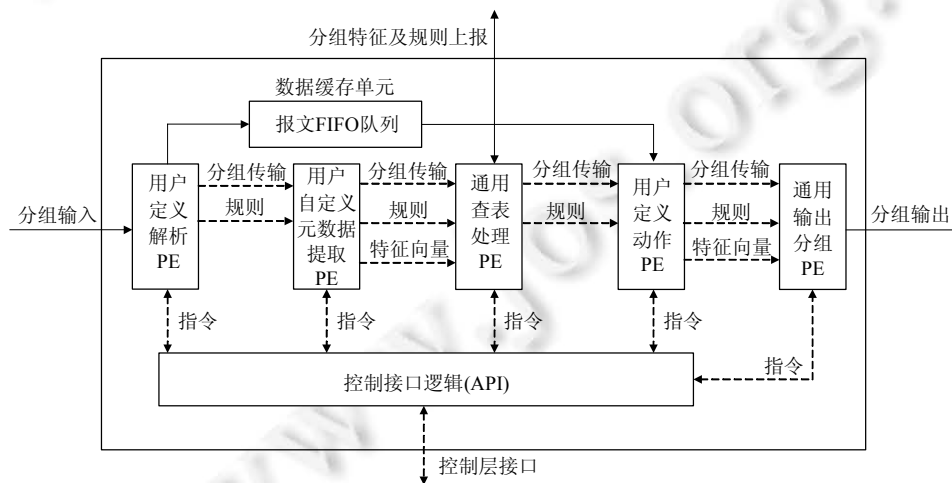


Fig.1 Pipeline structure of network processor

图 1 网络处理器流水线编排结构

学者谭章熹等人^[8]认为,网络处理器本身必须具备以下 5 点要求:(1) 拥有网络分组并行处理能力;(2) 具有高效的处理速度,能够达到分组的实时处理;(3) 拥有一定数目的网络专用协处理器;(4) 具有高度的可编程性和扩展性;(5) 能够快速投向市场,尽量减小再开发周期。

在这种条件下,除了拥有可靠的硬件架构外,网络处理器芯片内的各 PE 上需要执行相关程序进行单元间通信和转发表、提取元数据等操作。依据不同的网络处理任务,软件系统分为控制层面和数据层面:控制面搭载网络协议栈以及实时操作系统等进行管理和策略控制;数据层面包括控制程序、管理程序和转发程序等完成对数据的高速和定制化处理。

1.2 网络处理器发展以及挑战

网络处理器作为一种支持各项网络协议实现的特殊处理器,它具备一定的可编程性保证部分工作所需功能的动态加入^[9]。网络处理器问世于 1999 年,并即刻被网络设备厂商和半导体公司投资制作。随着学术界和业界对网络处理器的重视,不同体系结构、不同应用场景的网络处理器也相继出现。

作为推动下一代网络向灵活性和高性能结合的关键技术,新型网络处理器的体系结构研究首先成为了该领域的热点问题^[10]。为了使得网络处理器更具灵活性,以便对新型业务进行处理,中间网络设备开始不仅仅满足于基本的转发和查表操作。各类基于新型网络处理器体系架构的 QoS 路由器相继出现,如基于 DiffServ^[11]的 VERA(virtual extensible router architecture,虚拟可扩展路由器体系结构)可扩充路由器。这些设备主要关注可扩展性和灵活性,稍微兼顾了效率。随着对 QoS 控制策略的不断深入研究,网络处理器架构设计逐渐偏向可重点支持某一特定需求的协处理器设计^[12]。协处理器利用自身计算性能,增加了网络处理器针对特定应用的处理效率。当然,协处理器一定意义上拖累了网络处理器的灵活性,针对协处理器而设计的流水线或者体系架构更改成本较高。

为了解决上述问题,也随着网络环境的日益复杂,近年来,硬件领域以及编程语言的发展红利也逐渐映射到网络处理器研究上来。ASIC(application specific integrated circuit,专用集成电路)作为网络处理设备的初始应用芯片,在应用领域逐渐广泛并趋于灵活的网络市场上逐渐不被人满意。更多的半定制化芯片,如 FPGA(field programmable gate array,现场可编程逻辑门阵列)应用到了网络处理器设计中^[13]。网络处理器应用开发难度也影响着其灵活性利用新型的可编程技术,简化网络处理器设计的方式方法也成为研究热点。

网络处理器的发展也面临着对应的挑战。面向新型的网络体系结构,从数据网到电信网、从接入层到骨干层,网络处理器开始被广泛开展研究以及应用。随着需求的推动和技术间的交叉融合,网络处理器在平衡高速、灵活特性上逐渐出现矛盾^[14]。随着通用性强、开发环境友好、软硬件融合度高等设计理念都逐渐泛化为数据处理技术从而成为潮流,网络处理器面临着从交换、路由、网络融合与演进等传统业务拓展到适应新型体系结构、满足特殊需求型业务甚至应用于终端领域的重大挑战。

在这种大背景下,将网络处理器的“高性能”“高灵活”结合的研究局面出现,下一代网络处理器设计成为了研究热点。下一代网络处理器的核心设计方法是结合应用场景,将分组处理与转发的高性能和可编程、可演进的灵活性进行结合,使得其能够在自身资源属性的限制下,完成可定制、可重构的,并且提高搭载网络处理器设备的效率、可编程性、鲁棒性和安全性。目前,网络处理器的“下一代”设计思路主要体现在 3 个方面:利用新型可编程技术的下一代网络处理器、面向新型网络体系结构的下一代网络处理器、针对新型高性能业务的下一代网络处理器。具体的分类框架如图 2 所示。同时,为了使下一代网络处理器更具有有效性和可用性,工业界开始针对新型设计方法进行产品化、原型系统化的输出,本文也将会介绍这一部分内容。

本文将重点关注下一代网络处理器设计思想和体系结构,将这些虽然没有量产但是实现原型系统的网络处理器进行体系化的整理、介绍以及对比。

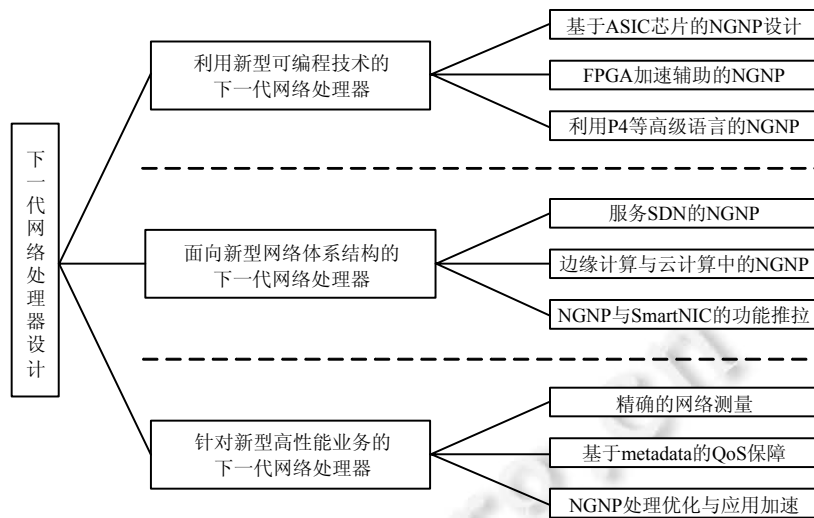


Fig.2 Classification framework of next generation network processor design methods

图2 下一代网络处理器设计方法分类框架

2 利用新型可编程技术的下一代网络处理器

网络处理器的灵活性依赖于两点:芯片的可编程性以及平台编程语言的交互难度和编写难度.芯片作为一个集成电路的载体,因其特有的体积小、重量轻、处理速度快等优势,得以在各项工业制造中广泛应用.可编程技术在不同的硬件上有不同的体现方式,由用户在使用中通过计算机指令选择不同通道和电路功能进行编程实现^[15].网络处理设备利用可编程芯片的起源较早,从开始的单片机到 ASIC、FPGA 等,除了处理网络协议,还可以扩大转发表,满足高性能需求.网络处理器作为一种特殊器件,其借鉴不同可编程芯片架构的融合开发,是下一代网络处理器原型的重点方向.在当前对中间网络设备处理需求多样、复杂的情况下,对芯片编程的语言可操作性,成为了评判网络处理器灵活性的一个重要方式.硬件描述语言随着 EDA(electronic design automation,电子设计自动化)技术的发展成为一种趋势.目前,最主要的硬件描述语言是 VHDL 和在 C 语言的基础上发展起来的 Verilog HDL.两者语法相对严格,主要关注数字电路和系统的设计与描述,功能虽然覆盖网络处理但是针对性不够,编写编译复杂.斯坦福大学的 McKeown 教授^[16]于 2014 年设计并提出了数据平面特定领域编程语言 P4,充分发挥了网络处理中数据平面的编程能力,也为网络处理器的设计提供了新的思路.

2.1 基于 ASIC 芯片的 NGNP 设计

网络处理器设计之初,ASIC,FPGA 以及 NP 作为不同的架构都可以搭载在路由器上,学术界也开始争论这几种芯片架构对交换机等设备的适用性.拥有不同特点的 3 种芯片的相互融合借鉴,是下一代网络处理器的设计潮流.2013 年,Teubner 和 Woods 编写了利用可编程门阵列 FPGA 进行网络数据处理的专著^[17],该书主要关注了网络处理器的硬件系统的可编程性,对利用 FPGA 进行网络数据流的处理进行了系统化的整理和指导,讨论了基于 ASIC 芯片对网络处理器的设计影响.

自 2005 年开始,3 种芯片的架构融合设计潮流兴起.David 和 Taylor 等人^[18]探讨了利用 FPGA 和 ASIC 等芯片形式,用以硬件辅助设计权衡的健壮的头压缩(robust header compression,简称 ROHC)方法.ROHC 在网络处理器上实现,并为分组交换网络中的无线通信提供更为有效的无线链路使用.Renterghem 等人^[19]研究了一种适合以太网接入节点的网络处理器,借鉴 ASIC 数据包处理器设计方法,通过多个处理器在多核环境下并行运行,并达到 10gbit/s 的以太网速率.该处理器有一个优化的体系结构来处理流处理任务,如解析、分类和包操作.VLIW(very long instruction word,超长指令字)指令集允许 ASIP(application specific instruction set processor,专用指令集处理器)内部各功能单元之间的高度并行,并有专用指令来加速典型的包处理任务.当然,上述两点研

究没有考虑可编程性,无法灵活更新任务.来自开罗大学的 Suleiman 等学者^[20]介绍了一种利用 ASIC 设计的嵌入式应用的低功耗、中性能网络处理器,它的典型时钟频率为 260MHz,功耗为 0.11mW/MHz,功率效率为 8.78dMIPS/mW,这使得它非常适合嵌入式和实时系统.但是实验验证,其处理性能不高.Yoon 等人^[21]提出了由多核处理器实现的一种适用于未来互联网和需要多层协议处理的高端网络服务的全流系统(OmniFlow system).作为一个基于流的网络处理器,使用了 65nmCOMS(complementary metal oxide semiconductor,互补金属氧化物半导体)技术,它包含 40 个处理核,并利用 ASIC 实现.OmniFlow system 可通过两个 SPI-4.2s 处理传入的包,并拥有 40Gb/s 包转发可编程引擎.文中也指出,该方法包含较多处理器核,如何进行并行调度也是未来的研究重点.

2.2 FPGA加速辅助的NGNP设计

Zhang 等学者^[22]提出了一种基于 FPGA 的高速 OBS(optical burst switch,光突发交换)核心节点 BCP 网络处理器,并提出了一种新的有效的波长调度算法.它不仅在调度有效波长时具有较高的带宽利用率,而且调度时间更短.当系统时钟为 200MHz 时,BCP 只需 180ns 的处理时间:约 50ns 用于调度有效波长,130ns 用于核心节点的其他问题.上述方法针对业务较为普通,获得高带宽后,面对拥塞情况没考虑.在用于特定业务的网络处理器方面,Niu 等人^[23]提出了一种基于 FPGA 的可配置的 IPSec 处理器,其本质是高性能在线网络安全处理器.它由两个嵌入式 32 位 CPU 核和一个 SoC 上的 IPSec 协议处理器构成,4 个并行 AH,ESP,AES,HMAC-SHA-1ip 核分别连接到 IPSec 处理器中的 8×8 交叉开关上,实现了 200MHz 时 1.5Gb/s 的吞吐量.虽然关注点是网络安全,但是进行网络流量可视化难度较高.Veerapraphatp 等人^[24]介绍了一种基于 FPGA 的网络处理器设计与实现,它是一种可扩展的分组交换结构,具有存储转发、错误管理、电源管理和安全等先进的网络功能.它不仅具有数据包化/去封装化、频率转换、数据大小转换等基本功能,还具有有限电路复杂度和较低的成本.然而,该方法在灵活性上的体现还是基于设计时的应用下发,没有考虑任务热部署,无法实现除预设能力外的更多功能.

由于预测网络流量以每年大约 60%的速率快速增长,因此迫切需要在 NGNP 上利用其他成熟芯片原理降低功耗.Roy^[25]讨论了 3 种减轻 NP 功耗的技术:异构多核架构、通用功能的硬件加速和非对称扩展技术. Shamani 等人^[26]介绍了一种基于 FPGA 的网络处理器的 COFFEE RISC(reduced instruction set computing,精简指令集计算机)协处理器,它能够通过控制器局域网(controller area network,简称 CAN)协议与外围设备通信.此外,核心和协处理器之间的数据传输在主数据总线上不引入任何工作负载,减少耗能的同时,最大可达 825MHz 的最大工作频率.上述方法由于性能上受限于设计架构,很难满足当代网络业务的高性能需求.

利用 FPGA 进行灵活与性能优化的 NP 研究方面,Gadre 等人^[27]提出了一种利用 RISC 处理器作为高速网络接口的、基于 FPGA 实现的网络处理器,其使用 RISC 处理器作为网络处理器的专有高速网络接口卡.这种 NP 可以进行多线程协议处理.文章还解释了在确定了完整的设计规范并在硬件上实现了整个设计之后,支持新协议所涉及的增量变化以及调试复杂协议处理和纠正协议实现错误所需的调试策略.Li 等人^[28]的文章介绍了一种基于 FPGA 加速平台的高柔性和高性能的 ClickNP,该处理器灵活性高,可以服务基于高级 C 类语言的可编程点击模块化路由器.其原型拥有每秒高达 2 亿个包的流量处理速度,并且具有低于 2 μ s 延迟.该方法由于设计性能要求高,加载策略过多,导致芯片产业化过于困难.

打破传统的芯片体系结构,将多种芯片优点混合设计进行网络处理芯片开发,成为 2018 年之后的主要研究热点.Zhou 等人^[29]设计并实现了 ECTCAT 实时应用的灵活快速路径可编程网络处理器.作为家庭网关的快速路径处理器,最多可用于 10GPON 访问.使用 VLIW(超长指令字)体系结构,可以并行运行多达 8 条指令.在 SMIC(中芯国际) 65nm 技术下,在 1.75Ghz 的运行频率下,最多可支持 7Gb/s 的数据吞吐量.同时,该处理器对于 1G 以太网具有最小 40ns 的端到端延迟,且无抖动.不过该设计受限于定制化服务,如何面对今后的网络发展是一个无法解决的问题.2019 年的 INFOCOM 上,Su 等人^[30]甚至提出了一种基于模板的网络处理器自动生成框架,通过将 P4 程序自动转换为 VHDL 程序来生成具有匹配动作结构的网络处理器.作为一种基于模板的解决方案,其使用 P4 程序自动转换为 VHDL 程序,从而在 FPGA 中生成包转发处理器,最后实例化生成网络处理器.该方法实例化过程复杂,适合学术界进行学术研究,不适合现场部署.

2.3 利用P4等高级语言的NGNP

一般地,转发芯片绝大部分都不是可编程芯片,无论是路由器中采用的 NP,还是交换机中采用的 Switch Chip,都不是可编程芯片^[31]。这些芯片的硬件转发逻辑已经设计好,无法通过调整软件参数去更改。编程语言的协议或者是平台相关性随着芯片种类的冗杂导致数据平面控制平面分离失效,网络处理器可编程性变差^[32]。P4的出现开始改变这种局面,网络处理器原型设计开始考虑对这种语言的适配性。

领域专用语言(domain-specific language,简称 DSL),如 P4,以其高生产率在网络领域得到了广泛的应用。然而,在 FPGA 上实现 P4 程序的自动转换是很困难的。

Cao 等人^[30]提出了一种基于模板的网络处理器自动生成框架,通过将 P4 程序自动转换为 VHDL 程序来生成具有匹配动作结构的网络处理器。该框架首先从 P4 程序中提取功能模块和相应的参数,这些功能模块将映射到预构建的高效 VHDL 模板以获得高性能。最后,所选模板将被实例化,并根据需要安排形成一个完整的处理器。实验表明:生成的处理器不仅具有较高的吞吐量,而且具有较低的延迟。该方法的缺陷是精度较差,测试压力强度不够。He 等人^[33]建议利用 P4 实现 NP 的网络功能虚拟化(NFV),可以处理与协议无关分组处理的 P4 语言,在计算和存储硬件方面等运营商指定的网络功能实例化具有很好的优势。同时,可以利用 P4 增强的网络硬件的性能改进。Cao 等人^[34]提出了一种将 P4 程序转换为 VHDL 语言并在 FPGA 平台上实现 NP 的框架,在该框架中,引入了一种基于匹配动作的硬件体系结构,其预构建的模板库用于编译,包括与特定的明确设计的组件相对应的优化 VHDL 模板。通过调用、配置、优化和实例化这些参数和关系,将它们映射到相应的模板,这样生成的 NP 占用资源少、吞吐率高、延迟低。当然,上述方法原型到芯片的过程比较困难。

NP 与 P4 的结合可以在灵活性上的质变,这首先给流量监测技术带来了新鲜血液。Martins 等人^[35]提出了利用 P4 实现的 NP 中搭载概率数据结构来监视多租户网络。文章为每个租户实现了 sketches 架构,利用位图和计数器数组两个概率结构来进行独立的网络监视。该方法没有考虑带内测量带来的精确问题,导致测量结果精确度不高。Paolucci 等人^[36]利用 P4 语言对 SDN(software defined network,软件定义网络)交换机的数据平面结构和行为进行编程。实例化的 NP 可以自定义管道和有状态对象,支持复杂的工作流、用户定义的协议/头和有限状态机强制。原型系统显示出良好的可扩展性性能和总体延迟。该方法没有进行现网测试,实际应用能力存在一定的疑问。在网络管理方面,Yazdinejad 等人^[37]设计了一种与静态网络设备不同的 NP 体系结构,它在数据平面具有高灵活性和可编程性,支持特定分组的解析和处理,原型系统采用了 P4 语言在 FPGA 上实现。Guan、Jiang、Benáček 等人^[38-40]都利用 P4 设计了负载均衡的网络监控、网络拥塞控制以及线速转发框架,使得下一代可能的异构网络硬件能够根据不同领域的特定应用程序的需要进行运行时定制。这些方法注重灵活性,在部署时间、处理性能两方面考虑不足。

P4 除了能够提高 NP 的灵活性,解决了有状态数据平面的局限性之外,也能利用存储和计算性能的改变,实现学习功能,提高性能。Sviridov 等人^[41]引入并提供了 LODGE 的设计准则。LODGE 是一个模型,根据该模型,分布式网络应用程序可以基于在其他交换机上共享的一些全局变量在每个交换机上利用 NP 做出本地决策。应用程序可以支持在 P4 和开放包处理器的有状态数据平面中学习并检测分布式拒绝服务(DDoS)攻击。Pontarelli 等人^[42]的目标是使 NP 进行预先操作,在保持高速多 Gb/s 操作的同时可编程。他们提出了一个特定领域的 NP 架构,称为包操作处理器(PMP),能够有效地支持微程序实现隧道、NAT(network address translation,网络地址转换)和 ARP(address resolution protocol,地址解析协议)应答生成等相关操作。上述方法由于需要离线学习,对数据要求严格,准确性过度依赖先验处理知识。

3 面向新型网络体系结构的下一代网络处理器

近年来,网络的应用场景变化复杂,传统网络体系结构在时延、传输性能上的弊端逐渐显现^[43]。新型的网络体系结构成为了学者们从根本上解决各种网络问题的研究热点,这种学术潮流也影响了学者们对网络处理器的设计模式。数据平面与控制平面分离原本就符合网络处理器的设计思想,面向软件定义网络这一新型网络体系结构进行 NP 设计,可以增强网络各节点处理性能,灵活开展各项业务。去中心化的下一代网络体系结构设计

思潮体现在了边缘计算中的网络应用.边缘计算节点对计算能力要求高,其功耗和成本必须维持在普通环境下的可接受的程度^[44].所以在边缘计算以及云计算、甚至雾计算的节点上,NP 可以成为一种专用的预处理器,在海量级的数据报文流量下,利用高效的包处理能力提炼出相关有效载荷,转发至计算节点的不同核心计算处理器进行数据计算或者相关处理.NP 作为一种功能芯片,实现非核心业务向智能网卡(smart network interface controller,简称 SmartNIC)的功能推拉与卸载,可以帮助 NP 提高利用率,增强网络的转发与处理性能.

3.1 服务SDN的NGNP

SDN 的设计虽然从控制层面进行了逻辑上的简化,但是实现这种下一代架构需要各个交换机拥有足够的计算与处理甚至是存储资源^[45].使用下一代网络处理器应用在交换设备上能够加速 SDN 体系结构的流表下发、带内遥测以及网络监管的性能.

NGNP 在 SDN 的数据中心建设上有得天独厚的优势.Brebner 等人^[46]探讨了利用 FPGA 实现的 NP 原型作为光网络和数据中心服务器之间的一种经济高效的中介,以满足下一代数据中心的需求.在数据中心本地或是在载波网络中,配备了这种 NP 的主要作用有 3 个:首先是从服务器上卸载网络处理功能,以便在较低的延迟下实现更高的吞吐量,同时释放服务器资源以集中于数据处理;第 2 个是为不同的通信协议和标准提供灵活的硬件支持;第 3 个是作为一种可管理的资源,它可以提供支持 SDN 和网络功能虚拟化(NFV)的软件控制器和管理程序所需的数据平面功能.该方法没有考虑突发流量情况下或者是网络攻击时的 NP 性能表现.为了破除兼容性带来的 OpenFlow 控制平面在许多网络平台上的部署障碍,Suñé 等人^[47]描述了 xDPd 和 OpenFlow 的网络处理器.硬件无关部分主要是使用修订的 OpenFLASH 库(RoFL),它为 OpenFLASH 控制器和数据通路元件的开发提供了基础,并允许为各种设备开发特定平台的驱动程序.Belter 等人^[48]描述了一种在软件定义的网络体系结构中实现 NP 的支持发现和 HSL 翻译功能,它将基于 OpenFlow 的 AFA 消息转换为 NP-3 网络处理器中的内存结构.NP-3 存储器结构通过 EZchip 提供的 EZdriver 访问,其内存包含一个带有流条目的结构、包匹配和操作的专有二进制编码.这两种方法在提供控制器命令编译支持时没有考虑 SDN 架构下流量的特征与分组处理特点,导致过多指令集翻译带来的性能下降.

将输入业务流与控制器发送的规则相匹配,是 SDN 业务转发的一个重要组成部分,其执行效率对网络性能有很大影响.Wijekoon 等人^[49]提出了一种高性能、低成本的 SDN 交换机流匹配体系结构,通过定制 NP 处理特定于 SDN 的业务.该 NP 由一个用于流匹配的专用单元和一个自定义处理器组成,这个处理器的 ISA 被设计用来加速 OpenFlow 和 SDN 相关的任务,比如向流表添加条目、处理相关报文等.其使用 32 位长指令,操作码保留 6 位,能够保持最大频率 150MHz,其流量匹配单元也可保证 250MHz 工作.但是该 NP 的功耗很高,设计相对复杂,很难大规模地在 SDN 网络中部署.Blaiech 等人^[50]提出了一种基于网络演算模型和博弈论算法的公平共享网络处理器资源分配策略,该策略根据虚拟节点的处理,动态地映射合适的资源.在 NP 的实现中,他们专注于在多个协处理器中根据 OpenFlow 转发模型来重新分配资源的包处理任务.

随着业务的不断更新,SDN 仍然寻求灵活的方式定义网络设备的行为,控制平面需要能够充分利用现代网络硬件不断增长的能力及其多样性,使得控制平面加快对网络变化的探测和响应^[51].Belter 等人^[52]提出了一种新的 NP 硬件抽象,其首要目标是公开 NP 和软件交换机的高级可编程能力;第 2 个目标是通过提供动态检查特定网络设备支持的功能的可能性,来扩展网络节点可编程性的概念;第 3 个目标是将使用新定义的 API (application programming interface,应用程序接口)的编程语言引入到不同类型网络设备的数据路径的可编程抽象中.Kaljic 等人^[53]提出了一种基于现 CPU+FPGA 技术的深度可编程 NP 混合结构,以克服 OpenFlow 在实现新协议和高级分组处理功能方面的局限性.通过对实现和实验评估,该 NP 将交换任务减少到简单的包流表查找,证明了采用混合的 FPGA/CPU 结构是可行的.这种方法的不足是在 CPU 和 FPGA 混合编排时忽略了缓存架构的设计,导致性能较差

在 SDN 中,OpenFlow 表查找可能需要检查 15 个包头字段.为了针对 SDN 这种下一代网络体系结构进行优化加速,Qu 等人^[54]提出了利用 NP 的包分类实现与有效的优化技术.在多核 GPP 上,他们使用并行程序线程并行化搜索和合并阶段.该方法强调性能,可能会加大 NP 中 Cache 的资源占用,反而导致 Cache 失效.Sun、Bi 等人^[55]

针对 SDN 数据平面提出了一种新的状态数据平面体系结构 SDPA。为 SDN 交换机设计了一个专用 NP,通过新的指令和状态表来管理状态信息,进而实现扩展的开放流协议来支持平面间通信。Wu 等人^[56]提出了一种支持 Openflow 协议的高扩展性 SDN 交换机的 NP 软硬件协同设计,为了在灵活性、能量效率和性能之间取得平衡,NP 核心是基于集成了 ARM 处理器和 FPGA 的嵌入式片上系统(SOC)平台来实现的。特别地,在所提出的交换架构中设计了专用高速信道,使得多个交换机可以互连在一起,成为一个堆叠的交换机。Li 等人^[57]提出了一种基于 Cavium 的开放式 vSwitch 实现,Cavium 平台就是一种多核 NP,它支持包的零拷贝,处理包的速度更快。这两种方法在高速通道的使用上容易引发堵塞和不一致性,风险系数较高。

安全性是 SDN 中最具挑战性的问题之一,网络安全应用程序通常需要以比 SDN 数据平面实现允许的更高级的方式分析和处理流量^[58]。NP 可以在 SDN 数据平面(即交换机)上对分组进行预处理,以初步确定其在网络中的行为。近期,区块链被用于安全传输网络中的交易和文件传递。Yazdinejad 等人^[59]根据区块链的安全特性和对数据处理功能的支持,提出了一种新的 SDN 分组解析器体系结构,称为区块链启用分组解析器(blockchain-enabled packet parser,简称 BPP),并进行了利用 FPGA 的 NP 原型实现。在他们提出的架构中,NP 利用一个基于多元相关方法的数学模型,用于从观察到的数据包流量中检测攻击,能够做到处理速度快、灵活性强、消耗资源较低。但是该方法没有对 NP 本身的架构加以解析和更改,没有突破原有 NP 架构在使能安全业务时的限制。

3.2 边缘计算与云计算中的NGNP

边缘计算、云计算节点最注重节点的数据处理能力^[44]。作为分导网络数据的重要“关卡”,网络处理器对网络数据的预先处理可以倍速增加计算节点的处理效能,甚至可以缓解整体网络的压力。

在边缘计算节点中,领域特定的局域网语言(如 Click)是捕获应用程序一致性的好方法,但网络处理器仍有许多其他机会来提供更符合应用程序需求的功能。Mihal、Andrew 等人^[60]提出了一个处理元素,它在实现 Click 应用程序任务时非常有效。其使用 Click 作为高级输入语言编程 PE 提供了一种高效的方法,并利用 FPGA 实现了为 IPv4 转发应用处理超过 8Gb/s 流量的 NP。Nayaka 等人^[61]介绍了一种用于片上系统(SoC)的以太网 NP 设计,该处理器实现了所有核心的分组处理功能,包括分组和重组、分组分类、路由和队列管理,提高了交换/路由性能,使其更适合下一代网络(next generation network,简称 NGN)的各项节点。该 NP 可支持 1/10/20/40/100 千兆链路,具有速度和性能优势。当然,其也具有片上系统的共同弱点,灵活性受限于定制流程。

加速 NP 中的数据流处理流程、增强节点性能、减少复杂度,是面向计算节点的 NP 设计重点。Ihor, Tchaikovsky^[62]成体系地介绍了面向边缘计算的处理器体系结构和服务,描述了网络处理器在设备流量处理和形成中的应用。他指出:在满足这些要求时,重点是包的处理速度。但是该文章没有讨论 NP 在遭遇计算节点流量突发时的情况。Kanada 等人^[63]提出了一种利用多个包处理器控制(packet-processing cores,简称 PPC)NP 中的分组处理的方法。通过这种方法,NP 芯片外的 CPU 对复杂的控制信息进行部分处理,并将其分为简化的控制包,这些控制包被发送到控制处理 PPC。利用 PPC 的数据交换机制(例如共享存储器或片上网络)来控制数据处理,这些机制比原有机制更加统一和简单。处理器关联性是提高 NP 性能的有效途径之一。He 等人^[64]详细分析了网络处理器的缓存特性,提出了一种兼顾负载均衡和数据包相关性的多核网络处理器调度算法 BLA。该方法尝试将同一个流的数据包调度到同一个 NP 核心,同时保持核心之间的工作负载平衡。但是增加核间调度,使得该方法牺牲了一定的性能。

链路速度的显著和连续增加以及应用的多样性,要求新的云计算节点服务高效又灵活。Niu 等人^[65]提出一种基于 NP 的高效流量管理 QoS 调度机制,详细讨论了利用数据平面软件体系结构的操作过程。他们将 NP 的一些协处理器设计得更加亲和计算节点业务,如准确的流量分类、灵活的访问控制和三步调度等。针对网络应用中数据量大、实时性要求高的问题,Wu 等人^[66]通过优化配置 NP 控制和数据平面,提出一种基于核心处理器的快速数据包处理的 NP 体系结构。它减少了 CPU 的调度,与平均分配 CPU 内核资源相比,提高了 30%的包转发率。上述方法的缺点是没有引入负载均衡考虑,准确的流量分类带来的高功耗,加大了计算节点的环境、能动压力。

与 DPDK(data plane development kit,数据平面开发套件)的设计模式类似,很多云计算或边缘计算节点不希望所有的数据通过之前搭载的网络协议栈进行处理,为 NP 专门实现一种协议栈,成为了计算节点的加速方式。

Tang 等人^[67]提出一种实时操作系统中的多径可靠数据传输(multi-path reliable data transfer,简称 MPRDT)系统,该系统在多核 NP 的实时操作系统(real time operating system,简称 RTOS)中实现了一个基于 UDP 的可靠数据传输栈,以加速网络处理器的可靠数据传输.该系统由两部分组成:基于 UDP 的可靠栈(URS)和连接管理模块.系统采用多径方法,使系统能够通过不同的接口管理到不同接收端的多径连接,充分利用网络接口和带宽资源.

随着服务器的虚拟化的发展趋势,出现了一种新的网络访问层,它由运行在服务器平台上的虚拟交换机组成,为同一物理服务器上的虚拟机(VM)提供连接.Blaiech 等人^[68]提出了一种策略,旨在通过将包处理任务扩展到 NP,以提高虚拟交换机的性能.该策略基于处理器资源的自适应动态分配,分配机制包括将虚拟交换机任务映射到足够的资源集,即多核数据路径或硬件加速器数据路径.传统的 NP 不能处理 7 层包,给计算节点带来了很大困扰,Bae 等人^[69]提出了一种新的 NP 互通结构,该结构通过将传统 NP 与通用处理器(GP)相结合,能够处理 OSI 第 2 层(L2)~第 7 层(L7)的数据包,并在不增加硬件开销的情况下提高 NP 中数据包处理的吞吐量和负载平衡.实现上述方法并不简单,兼容性是搭载新协议 NP 的重要问题,无法识别分组头部或者协议偏差导致识别错误时,NP 的后续处理不仅消耗 CPU 资源,而且导致下发的转发和处理任务全部失败.

随着云计算中网络即服务(network-as-a-service,简称 NaaS)的发展趋势,加速租户 NFs(network file system,网络文件系统)以满足性能要求也具有重要意义.然而,为了追求高性能,现有的工作如 AccelNet 被精心设计,以加速数据中心提供者的特定 NFS,这牺牲了快速部署新 NFs 的灵活性.Li 等人^[70]提出了一种可重构的 NP 流水线 DrawerPipe,它将数据包处理抽象为多个由同一接口连接的 drawer.开发人员可以轻松地将其与其他 NFs 共享,只需在适当的“抽屉”中加载核心应用程序逻辑,即可实现新的 NFs.此外,他们还提出了一个可编程模块索引机制,即 PMI,它可以以任何逻辑顺序连接“抽屉”,从而为不同的租户或流执行不同的 NFs.这种 NP 设计方案暂时无法大规模投入生产.Li 等人^[71]提出了一个并行包处理运行系统,并探讨了一种基于关联度的包调度算法,以提高 NP 负载平衡度并减少缓存丢失.在并行数据包处理系统中,由于牺牲了部分缓存,NP 任务分发器和调度器能够在负载均衡和缓存关联性之间达到较好的折衷.方法缺陷是租户单点故障时容易使得全部服务器 NP 性能宕机,在高并发情况下其效率有限,安全性也一般.

3.3 NGNP与SmartNIC的功能推拉

NP 的设计原型一般以板卡的形式实现,所以 NP 的设计阶段与 SmartNIC 的设计方法类似,呈现形式也相似.可以预见的是:面向复杂网络应用和对网络压力不同的各项应用,实现 NP 和 smartNIC 的功能推拉是必备的节点能力,很多研究者也在开始寻找两者的平衡点.

NP 的使用受到不断提高的灵活性和高性能的分组处理的需求的鼓舞.此外,适应性要求、产品差异化和缩短上市时间鼓励在网卡中使用网络处理器,而不是包括特定用途的硬件.Cascón 等人^[72]提出了一种利用 NP 的并行性来提高通信性能的网卡,遗憾的是,他们没有提出特有的框架,而是基于包括 16 个多线程处理内核和包处理的优化设计 Intel IXP28xx 网络处理器.通过使用 NIC 卸载和/或加载策略,利用不同的选项来优化主机中的通信路径.Sabin 等人^[73]提出的 SmartNIC 是一种用户可编程的 10GE NIC,可以满足 HPC(high performance computing,高性能计算机群)和数据中心社区的高性能网络需求.这种 SmartNIC 支持开发特定于应用程序的卸载引擎,以实现与 NP 的功能推拉.应用程序开发人员可以实现应用程序感知的卸载引擎,网络开发人员可以测试和开发网络协议卸载引擎,研究人员可以测试和开发新的卸载协议和中间件.但该方法没有明确卸载协议类型,适用范围有限.

从增加包处理性能的角度看,SmartNIC 从主机处理器上卸载网络功能,使其部分功能通过板卡或者是 NP 进行实现,是一个较优策略.Le 等人^[74]提出了一种广义的 SDN 控制的 NF 卸载结构(unifying host and smart NIC offload,简称 UNO).通过在主机中使用多个交换机,它可以透明地将动态选择的主机处理器的数据包处理功能卸载到 SmartNIC,同时保持数据中心范围的网络控制和管理平面不变.UNO 向 SDN 控制器公开单个虚拟控制平面,并在统一的虚拟管理平面后面隐藏动态 NF 卸载.这使得 UNO 能够最佳地利用主机和 SmartNIC 的组合包处理能力,并根据本地观察到的通信模式和资源消耗进行 NP 参与转发的本地优化,而无需中央控制器的参与.这种方法依赖于板卡 CPU 的计算性能,可能会无法正确地判断本地网络状态.Cornevaux-Juignet 等人^[75]考虑

了采用嵌入式现场 FPGA 的 SmartNIC 辅助 NP 进行处理的新解决方案,提出了一种混合体系结构来实现灵活的高性能流量取证.这项工作结合了硬件性能、高吞吐量和软件高灵活性,以实现超过 40Gb/s 的数据速率,同时可以通过参数在运行时热配置.Cerović 等人^[76]提出了一种将数据平面包处理卸载到具有并行处理能力的可编程硬件上的体系结构,因此,他们使用 MPPA 这种大规模并行 NP 阵列组建的 SmartNIC,它提供可用于数据包处理的 ODP(open distributed processing,开放式分布处理)API,并且可以建立一个全网格无阻塞的第 2 层网络.当然,这两种方法注重转发性能,包处理性能考虑不周.

实现与 SmartNIC 功能推拉的 NP 可以扩大自身的网络监控能力.Huang 等人^[77]设计并实现了一个基于多队列网卡和多核 NP 的数据流捕获系统,该系统充分利用了轮询技术、多队列技术和原始设备技术,大大提高了系统性能.2019 年的 SIGCOMM 上,Li 等人^[78]提出了一种新的高速拥塞控制机制 HPCC(high-precision clusion control),HPCC 利用网络遥测技术(INT)获得精确的链路负载信息,并精确控制通信量.他们用可编程的网卡(NICs)模仿 NP 来实现 HPCC.这种方法能够在避免拥塞的同时快速收敛,以利用空闲带宽,并能在网络队列中保持接近零的超低延迟.这种定制板卡的拥塞控制方法部署环境受限于大规模数据中心.在网络安全问题上,S. Miano 的文章^[79]旨在利用 SmartNICs 构建一个更高效的处理管道,并为特定用例(即减轻分布式拒绝服务(DDoS)攻击)的使用提供具体的方案.他们通过透明地卸载 SmartNIC 与 NP 中的 DDoS 缓解规则的一部分,实现了 XDP 灵活性在操作内核中的流量采样和聚合时的平衡组合,并具有基于硬件的过滤流量性能.由于依赖 DDoS 配置规则,缺乏在线流量学习功能,在真实网络环境中预计无法有效清洗 DDoS 流量.

SmartNIC 不仅可以包含 NP,还可以利用多项元器件加强网络包的转发或者处理性能.与此同时,学者将眼光转向保证 SmartNIC 与 NP 功能推拉结构的灵活性^[80].在 Caulfield 等人^[81]的文章中,他们重点讨论了基于 FPGA 的智能网卡(sNICs)和可编程交换机实现这一愿景的潜力.NP 和 SmartNIC 涵盖了从完全基于 CPU 的设计到完全定制的硬件的范围.基于 CPU 的可编程 NIC 提供一个或多个通用处理器,算法可以在这些处理器上运行.这种设计可以映射到 FPGA 上.这样可以保证推拉架构的可编程的能力,并且拥有完全定制硬件的效率和吞吐量特性.Microsoft 公司的 Firestone 和 Greenberg 等人^[82]提供了 Azure 加速网络(AccelNet)解决方案,其使用基于 FPGA 的自定义 Azure 智能网卡将主机网络卸载到硬件.利用 NP 的性能和 FPGA 的可编程性,为客户提供小于 15ms 的 TCP 延迟和 32Gb/s 吞吐量.这种方案在运行正常时缺乏日志记录,在业务演进过程中必须重新匹配软件定制方案,增加了运维成本.

4 针对新型高性能业务的下一代网络处理器

下一代高性能业务是指随着平台软硬件的迅速升级,在网络监测、网络传输、服务质量保证、VR(virtual reality,虚拟现实)以及高带宽低时延终端应用、网络安全以及物联网等方面新兴的复杂业务.这些业务一般需求更改时间快、性能要求高、针对性强,部署在当前的网络处理器中难度很高.针对下一代高性能业务,业界希望能够设计出符合要求的专用网络处理器满足相关需求.网络监测和网络传输的基本要求是,网络处理器具备精确网络测量能力^[83],测量过程的自适应性和精确性直接影响了监测结果和传输效率.利用网络处理器获得的数据包元数据(metadata)能够定制化地完成不同的服务质量保障.在性能方面,具备多核的网络处理器的核间调度、提高 Cache 亲和性以及分组向量化处理是当前研究的热点.

4.1 精确网络测量

网络测量作为众多网络管理应用的基础,其一直作为众多学术研究者的重要研究领域.网络处理器具有计算和存储能力后,借助报文的往返所携带的各项遥测数据进行非全域视角下的网络资源视图绘制,可以成为网络测量方向的发展重点.面向这一热点,NGNP 的设计对精确的网络测量进行了偏移.

高速捕获和处理流量中的数据包,是 NP 在精确网络测量中的重要应用.Ficara、Lu 以及 Li^[84-86]都利用了已有的网络处理器实现架构,如 Intel IXP2400 网络处理器 PCI-X 卡实现网络流量测量,并在此基础上实现了二维矩阵测量器和入侵监测系统等.这几种方法在思想上贴近下一代网络处理器架构设计,但是由于受限于已有架构,简单地通过下发基本测量任务使得其性能较差.Yang 等人^[87]提出了一种被动 HTTP 流量性能测量的 NP

架构,将对象分为不同的源/目的 IP 地址对,并使用对象间请求时间间隔来判断这些对象是否属于同一页面,从流量中实时地测量 HTTP 性能.该系统可以在高速网络中工作,可以部署在 ISP 上.Yuan 等人^[88]提出了用 FPGA 作为专用 NP 原型平台的 ProgME 的方法.ProgME 可以整合应用程序要求,调整自身缓存,以规避大量流带来的可扩展性挑战,并实现更好的应用程序感知准确性.其核心是基于流集的查询答案引擎协处理器,它可以由用户和应用程序通过提议的流集组成语言进行编程.上述文章基本实现的是 NP 中的协处理器,所以存在适用场景小、性能以及可靠性差的缺陷.

依靠 NP 的存储实现 NP 的智能测量任务下发以及测度值反馈分析,能极大地提高测量精度.Xie 等人^[89]提出了一种在 NP 上实现的新的动态测量方法(dynamic measurement method,简称 DMM),它通过记录在 NP 存储里的运行时计划操作和相应时间戳来导出测量路径;通过判断派生路径能否通过任务调度 IMC 模型接收;通过判断实际值是否满足初始状态的标签函数进行可靠性验证.当然,多核处理器进行测量值的传递一定是具有一定开销的,这种方法还是有优化空间.Ferkouss 等人^[90]提出了在 100Gb/s 混合 NP 上的 Openflow 多表流水线的记录及测量方案,该方案描述了几种将这些流水线查找表链接起来并将它们映射到不同类型的 NP 存储器设计方案.SDN 网络应用程序的要求可以在这种 NP 上灵活地实现,以便实现智能测量.但是,该方法没有使用 IXIA 硬件流量生成器以线速进行详尽的性能评估,这种 NP 的性能暂时未知.

平衡测量和转发的 CPU 负载以及 Cache、存储的利用率是针对下一代高性能测量任务的 NP 需要关注的重点.华盛顿大学的 Liu 等人^[91]研究如何使用 NP 的加速服务器在数据中心执行基于微服务的应用程序.文章提出了通过适当地将测量微服务卸载到 SmartNIC 的低功耗处理器上而不会造成延迟损失的负载均衡方法,这种方法依然面临网络流量路由和负载均衡、异构硬件上的微服务布局以及共享智能网卡资源的争夺等严重挑战.在利用测量结果进行拥塞控制方面,Narayan 等人^[92]提出将拥塞控制从数据路径转移到一个独立的 NP 代理中.这个必须同时提供一个表达性的拥塞控制 API 和一个规范,供数据路径设计者实现和部署.他们提出了一个用于拥塞控制的 API、数据路径原语和一个用户空间代理设计,该设计使用批处理方法与数据路径通信.但是这种方法的 NP 设计复杂,无论部署在端节点还是重要的中间路由设备上都不适合,定位不清晰.

4.2 基于 metadata 的 QoS 保障

服务质量保障在硬件上的实现一直是学术届的研究热点.在数据平面上获取分组的元数据后,对数据传输链路以及流量进行简单的阈值分析,对相关服务质量进行保障在网络处理器变得可行.

有效进行分组分类、保障特定业务的分组延迟和丢失,是 NGNP 在 QoS 保障上的重点研究方向.Avudaiammal 等人^[93]在基于通用 NP 架构上实现并验证了高速、低复杂度的基于启发式的专用分组分类(high speed packet classifier,简称 HASPC)机制可以执行多维分组分类.通过多位的 Trie 数据结构用于对地址前缀对的搜索,从而有效地聚合了协议和端口字段的搜索结果,并将该方法应用到 IXP 2400 上.但是他们没有提出通用分组分类高效方法,只对多媒体分组进行 QoS 保障.Park 和 Lee 等人^[94]提出了基于流的动态带宽控制方法的网络处理器体系结构.他们提出的 FDBC(flow-based dynamic bandwidth control,基于流的动态带宽控制)通过使用流分类、识别活动流并重新计算活动带宽,通过流量工程的方法屏蔽网络拓扑配置,以便在以太网上提供有效的 QoS.该方法的缺点是:当传入流量的总大小超过最大上游带宽时,无论服务类型或流的属性如何,HLS 都会丢弃流量.

面向 IP 网的传统服务,Li 等人^[95]在 IXP2400 架构上改进了一种传统的 ACL 算法,将 TCP 的 ASK 和 RST 添加到 1 级分类索引中,减少了规则冗余.通过微码实现的 IP QoS 体系结构,利用 CAR 处理器实现了带宽限制功能,满足数据线速转发,有效地保证多媒体服务的低延时需求.同样地,Saleem 等人^[96]利用 DiffServ 架构对 IXP2400 编程,添加缓存单元,以便在查找操作期间减少对 SRAM(static random access memory,静态随机存取存储器)的访问,使得总体速度提高并减少延迟.当然,两种方法基于 SDK3.1 开发,没有成体系结构地更改核心 NP 的架构.Nguyen 等人^[97]提出了一种针对 NP 的多模式完全可重新配置路由器.设计 NP 支持混合式分组交换体系结构,该体系结构可以在运行时进行动态重新配置,以在虫洞和虚拟直通交换方案之间进行交换.配备了 QoS 驱动的仲裁器保证了无需预留资源的吞吐量保证服务,基于优先级继承仲裁机制有效地利用了网络资源,具有

动态的期限可感知的重新路由机制.但是该方案成本较高,没有进行负载均衡,等待 CMOS 技术对其进行综合评估.

在拥塞控制的流量管理架构方面,Benacer 等人^[98]提出了一种支持 5G 传输的流量管理器架构.他们设计了基于 FPGA 的 NP 原型对传入流量(数据包)进行管制、调度、整形和排队的模型.流量管理以要满足每个流所允许的带宽配额并强制执行所需的 QoS 目标的方式,对要发送的数据包施加约束.方法实现在 Xilinx 板卡上,注重可编程性,通过流量发生器进行测试,其真实效果未知.Iqbal 等人^[99]重新设计了 NP 中的数据包调度方案,以在最小化乱序数据包的同时,提高网络处理器的吞吐量.其调度策略试图通过保持流局部性来维护数据包顺序,通过识别激进流来最大程度地减少流从一个核心到另一个核心的迁移,并在多个服务之间划分核心,以获得指令缓存局部性.此外,调度程序将基于哈希的设计扩展到了多服务路由器,其中,内核被动态分配给服务,以改善 I-Cache 局部性.该方案通过减少乱序包来获得吞吐的 QoS 性能,过多的调度设计增加了数据平面和控制平面的耦合.

NP 中,QoS 调度器的性能直接影响新 QoS 需求层出不穷背景下的 NP 性能和部分灵活性.Yu 等人^[100]在线卡中利用 NP 提出了基于 DVFS(dynamic voltage and frequency scaling,动态电压频率调整)的数据包 QoS 调度器设计方法,该方法使用队列长度(QL)和链接利用率来控制线卡中的执行速率.通过不同的频率缩放策略,保障了节省能源时的处理器性能下降,在一定意义上解决了功耗和 QoS 调度器的性能均衡.该方法基于预测队列长度,但在预测时只是根据阶段时间内的队列长度均值,很难应对 bufferbloat 情况,容易导致全方案失效.Paul 等人^[101]使用机器(深度)学习技术为智能 IP 路由器中的多核 NP 开发了 QoS 增强型智能调度程序,他们将 NP 每个内核都以利用率驱动的期限感知模式处理传入流量,并且使用学习算法,在运行时以智能方式动态地最小化负载不平衡,保持内核之间的稳态负载分配.该方法最大程度地减少每个内核的计算开销,获得更高的吞吐量、更低的平均等待时间值和 PLR(packet loss rate,丢包率).但是机器学习本身就有极大的开销,计算和存储资源在学习方法加载过程中会造成转发能力下降.由于 NP 本身架构受限,该方法与真正做到在线学习调度还有差距.

4.3 NP的处理优化以及应用加速

针对众多的高性能业务,网络处理器在设计架构上需要进行改进,加速分组的处理能力以及针对特定应用进行包分类后的处理优化.

提高 Cache 亲和性能够提高 NP 多核处理器的处理性能,许多多核处理器实时操作系统提供了通过设置相似性掩码来提高任务迁移到处理器指定子集的可能性.Bonifaci 等人^[102]提出了强任意处理器相似性调度的概念,利用层次(层流)亲和力和掩码的系统硬件拓扑结构,在最早截至优先(EDF)调度策略上实现对强大的分层处理器相似性的支持.该方法强调降低处理任务的时间复杂度,将性能提高到 $O(m^2)$ 左右.但是亲和性实现原理复杂,可调度性损失过大.Jang 等人^[103]提出了多核 NP 的新颖网络过程调度方案 MiAMI,其根据处理器缓存布局、通信强度和处理器负载来确定最佳处理器亲和性.该方法可以适应网络和处理器的动态负载,同时以最少的处理器资源需求充分利用网络带宽.在 Intel SMP Symmetrical Multi-Processing,对称多处理和 AMD NUMA(non uniform memory access architecture,非统一内存访问)服务器上,处理器利用率的有效性改善率分别达到 65%和 63%.但是,该方法的缺陷是没有考虑存储资源的 I/O 性能,扩展到外围设备的流程相对复杂.

在 NP 处理多媒体流服务、实时或者是高性能计算应用服务时,需要处理器周期多、提高时钟频率和微体系结构效率的困难.Ortiz 等人^[104]提出并分析了几种配置,以在 NP 可用的不同处理器核之间分配网络接口.该方法根据相应的通信任务与处理位置,优化存储不同数据结构的存储器的接近程度以及处理核特性之间的相关性.该方法使用多个内核加速给定连接的通信路径,利用多个内核同时处理属于相同或不同连接的数据包的补充.Hanforda 等人^[105]基于追踪高速 TCP 流网络设备瓶颈的方法,将协议处理效率定义为系统资源(例如 CPU 和缓存)计算量.在多核 NP 终端系统中,将网络中断,传输和应用程序处理接收过程分配、甚至绑定给相关亲和的特定处理器内核.但是这两种方法同样实现困难,算法复杂度过高.

NP 使用多个数据包处理元素提高数据包处理的并行性,是处理优化的重要研究点.OK 等人^[106]提出了一种用于具有多个分组 PE 的 NP 新序列保留分组调度器,使相同流的数据包由不同的 PE 并行处理,调度程序通过

利用预先估计的数据包来保留每个流的输出数据包的序列处理时间.该方法的缺陷是没有关注成本效益以及分组处理的准确性.Roy 等人^[107]提出了 16 核的类 NP 多核加速计算体系架构,其中的每个核均配备有专用硬件,可在每次硬件加速器调用时快速切换任务.控制器侦听到重点任务时,将任务抢占请求发送到内核,减少了实时任务的延迟.该方法利用优先级阈值化技术,避免了低级的任务和行头阻塞的延迟不确定性.遗憾的是,该方法芯片化过程较慢,仅仅用 28nm 技术制作,但是正在研发最新版本,市场前景光明.

减少 NP 内的队列长度,可以提高多数应用相应速度.Satheesh 等人^[108]提出了一个新的动态重新配置 NP 的波动流量排队系统,该方法使用 Kolmogorov 微分方程分析了 NP 中的动态可重配置排队模型,并获得了 PE 数量、队列长度、平均等待时间和重配置时间的上限.该方法能够动态地调节被使用的处理器数量,当队列增大到指定阈值时投入备用处理器,并在队列减小时动态减小.Avudaiammal^[93]的文章里也有这种设计思想,有效地聚合了协议和端口字段的搜索结果,以分配不同的处理器核.两个方法也是基于原有 NP 架构 IXP2400,很难大幅度提高性能.Steven 等人^[109]提出了一种基于数据流微体系结构的 NP 设计方法,能够解决资源虚拟化和数据包处理的并行性问题.当然,这项研究仍在进行中,如何将块级数据流、连接性和运算符涵盖进入这种 NP 体系结构是一个难点.表 2 是文中提到的不同 NGNP 设计方案优势及不足对比.

Table 2 Summary of designs of next generation network processor in existing study

表 2 现有下一代网络处理器设计方案总结

NGNP 类型	技术类型	相关技术	相关工作	优势	不足
利用新型可编程技术的 NGNP	基于 ASIC 芯片的 NGNP 设计	健壮报头压缩	David 等人 ^[18]	权衡芯片架构优点,提供更有效无线链路使用	没有考虑可编程性,无法灵活更新任务
		适合以太网接入节点的 NP	Van 等人 ^[19]	体系结构优化,VLIW 指令集允许各 PE 高度并行	
		低功耗 ASIC 网络处理器	Suleiman 等人 ^[20]	适合嵌入式和实时系统	性能较差,实时处理滞后
		全流系统	Yoon 等人 ^[21]	适用于需要多层协议处理的高端网络服务	包含较多处理器核,并行调度开销极大
	FPGA 加速辅助的 NGNP	高速核心节点 BCP 处理器波长调度算法	Zhang 等人 ^[22]	调度有效波长时具有较高的带宽利用率,而且调度时间更短.	针对业务普通,获得高带宽后面对拥塞情况没考虑
		基于 FPGA 的 IPSec 安全 NP	Niu 等人 ^[23]	可提供在线的网络安全方案,实现了高吞吐率	网络流量分析难度较高
		可扩展的低功耗通用 NP 交换架构	Veerapraphap 等人 ^[24] Roy 等人 ^[25] Shamani 等人 ^[26]	灵活性高,具有存储转发、错误管理、电源管理和安全等功能	没有考虑任务热部署,无法实现除预设能力外的更多功能
		基于 FPGA 的高性能 NP	Gadre 等人 ^[27] Li 等人 ^[28] Zhou 等人 ^[29]	进行多线程协议处理,拥有每秒高达 2 亿个包的流量处理速度,并且具有低于 2 μ s 延迟	芯片实现过于困难,受限于定制化服务,无法应对新型业务
		基于模板的 NP 自动生成框架	Su 等人 ^[30]	使用 P4 程序自动转换为 VHDL 程序,从而在 FPGA 中生成处理器	测试压力强度不够,分组处理精度差
	利用 P4 等高级语言的 NGNP	协议无关的分组处理 NP	He 等人 ^[33] Cao 等人 ^[34]	利用 P4 实现 NP 的网络功能虚拟化,占用资源少、吞吐量高、延迟低	原型到芯片产业化的过程困难
		可编程的流量监测 NP	Martins 等人 ^[35]	为每个租户实现了 sketches 架构,利用多概率结构来进行独立的网络监视	测量结果精确度不高
		高灵活性多任务 NP	Paolucci 等人 ^[36] Yazdinejad 等人 ^[37] Guan 等人 ^[38] Jiang 等人 ^[39] Benáček 等人 ^[40]	在数据平面具有高灵活性和可编程性;支持特定分组的解析和处理;根据不同领域的特定应用需要进行定制任务	部署时间缺乏测试,导致其热部署任务时分组处理失效率高
		基于学习的预决策 NP	Sviridov 等人 ^[41] Pontarelli 等人 ^[42]	用 NP 做出本地决策,进行预先操作,在保持高速多操作的同时可编程.	对学习数据要求严格,导致处理分组的准确性差

Table 2 Summary of designs of next generation network processor in existing study (Continued 1)

表 2 现有下一代网络处理器设计方案总结(续 1)

NGNP 类型	技术类型	相关技术	相关工作	优势	不足
面向新型网络体系结构的 NGNP	服务 SDN 的 NGNP	SDN 数据中心的高效中介 NP	Brebner 等人 ^[46]	为不同的通信协议和标准提供灵活的硬件支持;提供支持组件控制器和数据平面功能	没有考虑突发流量情况下或者是网络攻击时的 NP 性能表现
		在 OpenFlow 控制平面上易部署 NP	Suñé 等人 ^[47] Belter 等人 ^[48]	为控制器和数据通路元件的开发提供了基础并允许为各种设备开发特定平台的驱动程序	没有考虑 SDN 架构下流量的特征与分组处理特点,导致过多指令集翻译带来的性能下降
		高性能 SDN 交换机流匹配 NP	Wijekoon 等人 ^[49] Blaiech 等人 ^[50]	动态地映射合适的资源,性能较高且保证成本低	专注于协处理器的使用,设计复杂,功耗较高
		高可编程性灵活 SDN 交换机 NP	Belter 等人 ^[52] Kaljic 等人 ^[53]	克服 OpenFlow 实现新协议和高级分组处理功能方面的局限性,使用新定义的 API 引入到不同类型网络设备的数据路径可编程抽象中	忽略了缓存架构的设计,无法进行流表特征学习,性能有待提高
		包分类加速 NP	Qu 等人 ^[54] Sun 等人 ^[55] Wu 等人 ^[56]	在灵活性、能量效率和性能之间取得平衡;具备专用高速信道,支持包的零拷贝,处理速度更快	会加大 NP 中 Cache 的资源占用,易引发高速信道堵塞和不一致性,风险系数较高
		安全的 SDN 交换机 NP	Yazdinejad 等人 ^[59]	从观察到的数据包流量中检测攻击,能够做到处理速度快、灵活性强、消耗资源较低	没有对 NP 本身架构加以解析和更改,没有突破原有 NP 架构在使能安全业务时的限制
	边缘计算与云计算中的 NGNP	流处理加速 NP	Mihal 等人 ^[60] Nayaka 等人 ^[61] Tchaikovskiy 等人 ^[62] Kanada 等人 ^[63] He 等人 ^[64]	增强节点性能,减少 NP 处理时间,支持高带宽链路,流内数据包调度算法效率高	没有考虑 NP 在遭遇计算节点流量突发的情况,频繁的核间调度对性能有一定的影响
		面向云计算节点多样化需求的 NP	Niu 等人 ^[65] Wu 等人 ^[66]	处理器亲和程度高,减少 CPU 的调度,整体提升转发率	功耗高,环境动力压力大
		搭载新型处理协议的 NP	Tang 等人 ^[67] Blaiech 等人 ^[68] Bae 等人 ^[69]	实现一种协议栈,处理器资源的自适应动态分配,不增加硬件开销的情况下提高 NP 中数据包处理的吞吐量和负载均衡	兼容性差,有无法识别分组头部元数据或者识别错误的可能,导致下发任务失败
		租户性能定向加速的 NP	Li 等人 ^[70] Li 等人 ^[71]	利用可编程模块索引机制,为不同租户提供不同服务,减少缓存丢失,折衷负载均衡性	在高并发情况下其效率有限,安全性也一般
	NGNP 与 SmartNIC 的功能推拉	支持 NP 功能卸载智能网卡	Sabin 等人 ^[73] Le 等人 ^[74] Cerović 等 ^[76] Huang 等人 ^[77] Li 等人 ^[78] Miano ^[79]	利用混合结构实现高性能流量取证,将 NP 功能卸载到网卡中,提高性能的同时保证了高灵活性	架构设计复杂,虽然实现了高性能和高灵活性的平衡,但是本质上没有对 NP 的架构进行更改,是一种过渡性质的技术
		灵活性提升的功能推拉架构	Caulfield 等人 ^[81] Firestone 等人 ^[82]	利用多项元器件加强网络包的转发或者处理性能,利用 NP 的性能和 FPGA 的可编程性为客户提供低延时高吞吐服务	在业务演进过程中必须重新匹配软件定制方案,增加了运维成本

Table 2 Summary of designs of next generation network processor in existing study (Continued 2)**表 2** 现有下一代网络处理器设计方案总结(续 2)

NGNP 类型	技术类型	相关技术	相关工作	优势	不足
针对新型高性能业务的 NNGNP	精确的网络测量	支持高速捕获处理分组的被动测量 NP 架构	Yang 等人 ^[87] Yuan 等人 ^[88]	可在高速网络中工作,整合应用程序要求,规避大量流带来的可扩展性挑战,并实现更好的应用程序感知准确性	适用场景小、性能以及可靠性差的缺陷
		利用 NP 的高精度测量方法	Xie 等人 ^[89] El Ferkouss 等人 ^[90]	精确性高,通过缓存进行准确的测量任务推导及下发	计算开销很大,缺乏线速情况下的性能评估
		负载均衡的专用网络测量 NP	Liu 等人 ^[91] Narayan 等人 ^[92]	增强了 NP 的实用性	资源争夺问题严重,异构流量测量问题考虑不周
	基于 metadata 的 QoS 保障	保障低时延的 NP	Avudaiammal 等人 ^[93] Lee 等人 ^[94] Li 等人 ^[95] Saleem 等人 ^[96] Nguyen 等人 ^[97]	保障分组交互的低时延,减少规则冗余,提升速度的同时减少了分组丢失情况	没有进行负载均衡,成本比较高,有效性有待提高
		流量可控与数据包 QoS 调度架构	Benacer 等人 ^[98] Iqbal 等人 ^[99]	满足每个流所允许的带宽配额,强制执行所需的 QoS 目标的方式对要发送的数据包施加约束	性能上受限于设计架构,维护表项代价高
		基于元数据学习的 NP 内分组调度	Yu 等人 ^[100] Paul 等人 ^[101]	引入机器学习方法,精准度高,功耗较低	开销大,计算和存储资源在学习方法加载过程中会造成转发能力下降
	NGNP 处理优化与应用加速	高亲和性多核处理优化	Jang 等人 ^[103]	可以适应网络和处理器的动态负载,同时以最少的处理器资源需求充分利用网络带宽	没有考虑存储资源的 I/O 性能,扩展到外围设备的流程相对复杂
		面向应用的服务加速	Ortiz 等人 ^[104] Hanforda 等人 ^[105]	优化存储不同数据结构的存储器的接近程度以及处理核特性之间的相关性	算法复杂度过高
		多核并行性处理优化	OK 等人 ^[106] Roy 等人 ^[107]	提高 PE 处理并行性,在每次硬件加速器调用时快速切换任务,避免了低级的任务和行头阻塞的延迟不确定性	芯片化过程较慢,仅仅用 28nm 技术制作
		处理队列减少的 NP 设计方案	Satheesh 等人 ^[108] Avudaiammal 等人 ^[93] Steven 等人 ^[109]	减少分组队列长度,动态使用处理器核数量,性能提高	没有将块级数据流、连接性和运算符涵盖到这种体系架构

5 下一代网络处理器的工业化及评测

5.1 网络处理器的工业化

国际上,作为最早的网络处理器生产厂家之一, Motorola 于 2000 年生产了面向中低端应用的网络处理器 C5,其只能执行第 7 层以下的分组分类作业.虽然能力较弱,但作为一种独立网络转发处理芯片,其出现代表着 NP 开始了工业化之路. IBM 于 2002 年生产了 PowerNP NP4GS3,其拥有 16 个协议处理器、7 个专业协处理器和一个 PowerPC 核心处理器,具有 2.5Gb/s 的报文处理能力.同样作为一款廉价处理器,其占据了国际国内大部分市场,但处理性能依旧不佳. 2005 年, Intel 推出了 IXP42X,其逐渐开始争夺业界主流位置,这款产品的处理单元内继承了数十个以太网口,内置了加速功能降低系统成本.这几款产品出现的年代网络业务简单,智能终端尚未普及,所以其灵活性重视不够,性能上差强人意.

随着网络技术创新浪潮出现,传统网络处理器开始顾此失彼,逐渐被市场淘汰^[110]. 2013 年, Cisco 推出了 nPower X1,其具备高可扩展性,也是首款支持 400Gb/s 吞吐率的单芯片,引领了利用新型可编程技术的网络处理

器产业化趋势。创新性地,其为软件定义网络构建,支持在运行中重新编程,大幅简化了网络运营。2015年,Mellanox公司发布了NP-5网络处理器,作为一款240Gb/s线速网络处理器,其峰值处理数据路径和CoS分类高达480Gb/s。同年,ExpressPlus推出了Juniper处理器,每秒可执行超过15亿次过滤操作,并可扩展到500Gb/s(1Tb/s半双工)。作为一款面向新型网络体系结构的下一代网络处理器,其利用3D内存的多级缓存架构,与以前的解决方案相比,物理占用空间减少了20倍,优化了功耗和空间要求。Nokia在2017年推出了世界首款可支持3.0Tbit/s转发的网络处理器,利用网络元数据作为安全解决方案的一部分,是针对新型高性能业务网络处理的产品代表之一;同年,Cisco设计了可支持400Gb/s转发的多核网络处理器,它包含672个处理器核心,拥有着大于6.5Tb/s的核心I/O带宽,外部DRAM(dynamic random access memory,动态随机存取存储器)用于大型数据结构 and 数据包缓冲。2020年初,收购了Freescale的NXP公司推出了S32G,该设备是第一款将具有ASILD安全性和网络加速功能(micro control unit,简称MCU)集成车载网络处理器,并且具有分组转发引擎用于以太网加速功能。当然,这几款产品设计与生产成本较高,市场化进程缓慢,大多在自家高端路由器或者交换机产品中应用。

中国的网络通信设备制造商在2016年以前以采购国外网络处理器芯片为主,以便获得硬件发展红利,将创新与产能焦点集中在网络业务出新和5G通信网络研究与建设上。然而随着中美贸易争端开始,2018年4月16日,美国商务部发布公告,声称未来7年内禁止中兴向美国企业购买包含网络处理器芯片在内的“敏感”设备,该事件直至同年7月12日才以中兴向美国支付近10亿美元告一段落。然而一年后,2019年5月15日,美国商务部将华为和其下属子公司列入出口管制名单,华为产业下的多款中高端路由器产品网络处理器芯片断供。中国业界及学术界逐渐认识到网络处理器芯片作为一种重要网络设备元器件,其设计与生产应当至少具备自主可控能力,在特殊时期保证网络处理器芯片的供应。

2019年,华为和中兴都开始进行自主可控的下一代网络处理器研发工作,设计方案的基本要求是,使得网络处理器具备高性能、低时延、可编程甚至是全流可视化。华为于2019年初推出了Solar S下一代网络处理器设计,其可以灵活地利用高级编程语言下发业务,保障低时延的同时,利用特有的存储结构,将流量的特征信息在时间域内存储,以便实时网络管理和精确测量,并且大多数开始应用于华为的高端路由器产品。同年,中兴发布了一款拥有先进内核互联结构、大容量微码指令空间、层次化的流量管理的100Gb/s网络处理器。在此基础上,设计了这款处理器的迭代更新版本原型系统,升级内部体系架构,利用多核的性能提升转发速度。这两款国产化网络处理器与国际市场主流处理器相比在性能上相差无异,但是在芯片工艺、产品能耗上还有差距,亟待弥补。

5.2 部分主流网络处理器的性能评测

一般地,对网络处理器分为4个层次,即硬件指令层、模块任务层、应用功能层和系统平台层来进行性能评估。通过在每一个层次上逐步分析,可将多级并行的系统性能评估问题转化为简单的串行程序性能分析,或者是简单的并行程序性能分析^[11]。在这种评估模型的基础上,利用网络测试设备搭建相关拓扑,连接测评仪器、控制器以及NP芯片(原型系统)开始测评被测设计性能。对于芯片数据包的处理能力的评测,一般采用M/M/1排队模型进行评估,利用流量生成器完成流量生成。接口驱动模块负责将数据流发送给被测NP。NP将处理后的数据发送给平台的接口响应模块。最后,接口响应模块将数据发送给相应评测模块,模块内置了相关算法对正确性和性能进行统计评测。

现有的网络测试设备厂商一般有思博伦、安捷伦、IXIA等。一般地,NP芯片的性能测试组网方式如图3所示。图3中,PC机通过串行COM接口对网络处理器进行配置和管理。其硬件编译环境通过JTAG(joint test action group,联合测试工作组)接口与网络处理器相连进行调试诊断。测试仪产生、发出以及接收流量也由PC机进行控制,测试仪与网络处理器原型之间通过万兆Ethernet接口进行流量传输、测试。针对上述产品,本文搭建了相关测试平台,购买并测试了C5,NP4GS3和IXP42X这3款NP的吞吐率以及处理时延。吞吐率为测试仪线速发送测试报文得到的速率统计信息,时延为利用测试仪时间统计功能得到的报文从发送到返回所消耗时间的一半。对于刚刚上市、价格昂贵无法购买的下一代网络处理器,本文通过查询企业产品手册、相关文献等对结果进行了统计。测试与统计结果展示于表3。

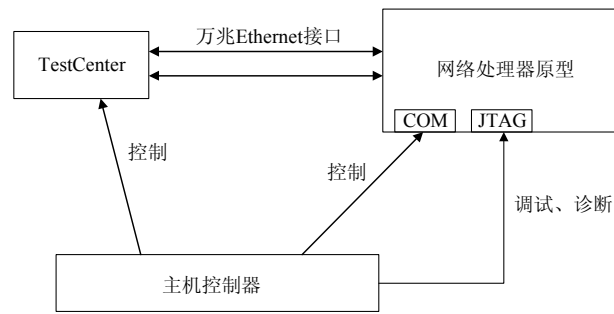


Fig.3 NP testbed

图3 NP性能测试床

Table3 Performance of some network processor products

表3 部分网络处理器产品性能

型号	所属公司	处理/转发性能	处理时延	灵活性	功耗	芯片工艺
C5	Motorola	17个可编程RISC内核;可执行第七层以下的分组分类作业,速度为5Gb/s	13 μ s	C/C++可编程;标准指令集	10.5W~23W	838个引脚 BGA封装
PowerNP NP4GS3	IBM	16个可编程微处理器;处理速度为2.5Gb/s	11 μ s	可编程,支持自定义功能,软件工具支持在不改变硬件可用性的情况下现场部署升级	1.8V电源电压	IBM SA-27E,0.18 μ m技术;1088针底面冶金-带815信号输入/输出的陶瓷柱网阵列(BSM-CCGA)封装
IXP42X	Intel	Intel Xscale内核,3个网络处理器引擎;处理速度约为100Mb/s	8 μ s	可编程;开发平台提供一致的开发环境,可以进行快速的产品开发和性能评估.IXP42X产品系列成员之间的引脚具有兼容性	1.0W~1.5W (工作在533MHz)	0.18 μ m工艺; 492-Ball PBGA封装
nPower X1	Cisco	具有336个报文处理引擎,每个报文处理引擎可以运行2个线程;处理速度约为400Gb/s吞吐率	约8 μ s	可扩展至多TB网络性能;专为SDN构建,支持在运行中重新编程,大幅简化网络运营	每比特能耗是思科之前产品的1/4	超过40亿个晶体管
NP-5	Mellanox	转发处理速度约为240Gb/s	-	集成流量管理功能,支持灵活地分组处理和细粒度的带宽控制	支持电源管理以最小化线卡和系统功耗;使用DDR3 DRAM以最小化功耗和成本	28nm工艺制造
ExpressPlus	Juniper	每秒可执行超过15亿次过滤操作	-	可扩展到500Gb/s(1Tb/s半双工);可提供完整的IP路由和MPLS,SDN的可编程性;易于部署	优化了功耗	28nm工艺并利用3D内存架构,物理占用空间减少了20倍
FP4	Nokia	转发速度约为3.0Tb/s;支持通道约为1Tb/s畅通流量	-	基于FP4的平台可灵活支持高密度的400GE接口和千兆速度链路;支持大规模的高级深度数据包查找和路由控制	与前几代硅片相比,FP4可以使每千兆位的功耗减少50%以上	结合了智能存储器和高度的硅集成,简化了电路板设计

Table3 Performance of some network processor products (Continued)

表 3 部分网络处理器产品性能(续)

型号	所属公司	处理/转发性能	处理时延	灵活性	功耗	芯片工艺
400Gb/s 多核 NP	Cisco	673 个通用 处理器,全双工 处理速度 400Gb/s	-	支持 C 语言和 汇编语言编程, 支持传统堆栈	大部分逻辑 工作中 760MHz 或 1GHz	22nm 制程,92 亿个 晶体管,643mm ² 裸片,混合 COT 设计流程
100G 下一代 网络处理器	中兴	处理能力为 100Gb/s	10μs	按需定制, 快速反应	-	-
Solar S	华为	转发能力为 3.6Tb/s	-	提供开放 APIs 编程;全流可视	业界平均 水平的一半	-
S32G	NXP	采用了 Quad Arm Cortex-A53 内核和 Arm Neon 技术;可用 于实时应用,安全操 作系统和 ECU 整合	-	可以通过更高级的操作 系统和更大的内存实现 更高级别的安全性能; 支持下一代 ADAS 应用 程序,同时提供通信功能	-	AEC-100 2 级 设备(-40℃~ 105℃)

6 未来发展趋势

尽管目前的网络处理器都考虑了可编程性和处理转发速度,但是没有在顶层设计上兼顾灵活和性能^[112].网络处理器可编程性多是借助开发平台或者是使用语言的先天属性,在此基础上调整相关数据平面控制平面的处理架构、元器件编排方式以及业务处理流水等,导致灵活和性能总是无法收敛到最优程度^[113].

6.1 高性能可演进下一代网络处理器架构

为了解决上述问题,下一代网络处理器应当朝着高性能、可演进的融合架构方向发展.针对处理优化和应用加速等高性能要求,融合架构的研究应当包括多级分组缓存管理框架、CPU 分组调度方法、可动态重构的应用加速等方面.一方面优化改进网络处理器基础设计架构;另一方面,则面向业务应用的加速进行实现^[114],并为上层提供相应的基础开发接口.在这种架构下,网络处理器的接口线速和转发能力都被良好地设计和实现,融合架构的可编程性、可演进性主要体现在硬件资源管理和可扩展 API 编程接口的研究.基于上述优化加速的平台架构,下一代网络处理器应当为用户提供高效、良定义、可扩展的 API 接口和开发模型,支撑上层业务应用的开发.提供平台无关的 API 接口、支持高级语言编程不仅方便了业务应用开发周期,也使得所设计的网络处理器的场景适应能力增强.在当前网络技术和人工智能结合的发展热点下,融合架构能够拥有足够的资源将部分学习方法卸载进入网络处理器,细粒度、高效率地实现包转发处理.

现有的网络处理器通常基于多 CPU 核加硬件协处理器阵列架构实现,加解密、查表等协处理器以 ASIC 电路形式实现,可以有效提升网络处理器进行分组转发处理的性能.然而随着硬件协处理器开发周期长^[115],且功能固定,难以满足网络处理可演化性需求以及面向多业务场景的应用需求.综上,本文提出了基于通用多核 CPU 加 FPGA 的高性能可演进下一代网络处理器(high performance evolvable network processor,简称 HPENP)架构. HPENP 设计的软硬件协同可扩展分组处理架构一方面利用通用多核 CPU 的可编程灵活性和并发处理能力,支撑网络分组深度处理^[116];另一方面,利用 FPGA 具有的硬件可重构特性,支持数据平面处理及加速模块的按需部署^[117].HPENP 具有转发模块与硬件加速模块的混合编排架构,可在此基础上实现基于模块 ID 映射表的流水线功能动态扩展方法,有效支持软硬件功能的动态加载,提升网络处理器面向多场景的应用能力和未来新型业务协议的演化能力.

6.2 软硬件协同分组处理流水线

高性能可演进的下一代网络处理器体系结构基于通用多核处理器及可重构 FPGA 硬件构建,其中的设计重点是实现软硬件协同的分组处理流水线. HPENP 支持 CPU+FPGA 平台软硬件高效数据通信,并且利用转发模块与硬件加速模块的混合编排方法.设计采用可编程硬件流水线、硬件流水线扩展、软件模块扩展以及软硬件协同扩展这 4 种方式提升网络处理器可编程及可演进能力,满足网络设备功能扩展的性能和灵活性要求.

如图 4 所示,软硬件流水线均包含多个在 CPU 上可按需动态加载的扩展模块,实现对不同处理功能的扩展.动态扩展模块一方面可以实现通用路由交换处理的“慢速路径”功能(如分片重组),也可扩展实现一些新的如按需部署的安全防护网络功能.可定制处理模块还可将目的 IP 地址是本地的分组发送到控制平面处理.动态扩展模块和控制平面软件通过网络接口发送分组时,该分组首先会送给 FPGA 中由可定制处理模块组成的硬件流水线处理.软件发出的分组可以对硬件流水线进行重入,进一步增加了下一代网络处理器处理的灵活性.由国产 CPU 内搭载的协议栈、驱动以及内核等通过良定义接口实现用户空间的交互,完成扩展性能.

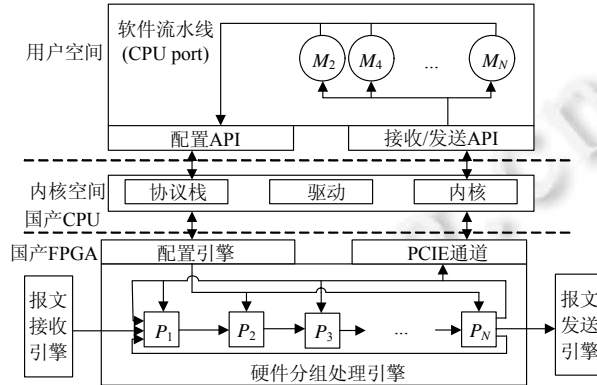


Fig.4 HPENP's architecture

图 4 HPENP 体系架构

6.3 多级缓存与分组调度

HPENP 架构实现 Cache、内存和外存相结合的多层次分组缓存管理,以打破“存储墙”问题对网络处理器系统性能瓶颈的影响.网络流量到缓存的调度与映射优化算法、多层次分组缓存动态管理与调度机制融入到了本架构设计中.借鉴计算机体系架构的多级缓存设计模型^[118],基于 CPU Cache、内存、FPGA 内部 RAM 和 DDR 外存设计多层次存储框架实现多级缓存的分组统一管理调度.针对突发数据流的可导致拥塞状况,HPENP 具备动态缓存分配方案,引用大容量 DRAM 作为备用队列缓存存储溢出流量.

HPENP 支持基于多核分组路径感知的分组调度技术,如图 5 所示.

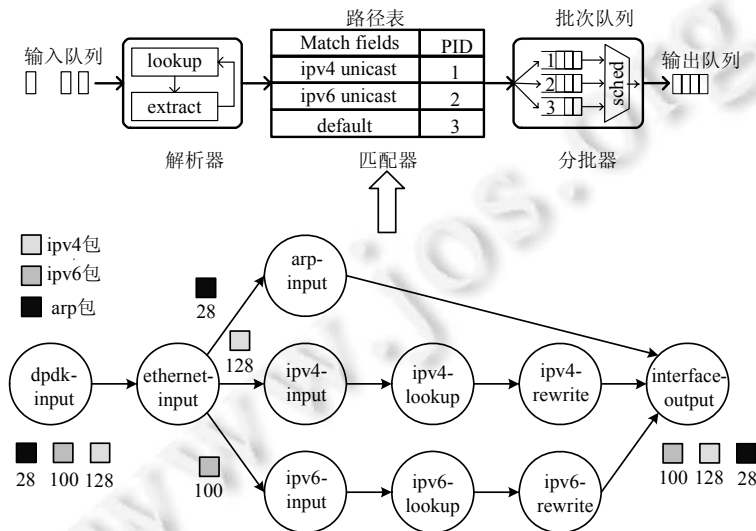


Fig.5 HPENP's packet scheduling model

图 5 HPENP 分组调度模型

分组调度感知器按分组特征预先分类进入 CPU 的分组,生成无处理分支的报文向量,以减少 CPU TLB (translation lookaside buffer,转译后备缓冲器)和 cache 失效.输入队列经过解析器、匹配器以及分配器进行分组的路径处理,以生成输出队列.各个硬件处理单元通过多项用户定义处理功能,降低分组 IO 过程中的性能损失,提升并行化处理性能.

6.4 编程接口及进程实现

HPENP 将 CPU 核、FPGA 逻辑资源、内外部存储资源进行统一抽象,并支持按需配置和动态分配,完成分组业务应用到系统各类资源的重构、映射与调度.HPENP 资源管理与编程接口主要研究层次化的资源管理和可扩展 API 编程接口,基于优化的网络处理器平台架构,为用户提供可扩展 API 接口和开发模型,支撑上层业务应用的开发.分层处理结构分为内核态与用户态两个部分,每一部分都是一个完整的开发环境,分别包含内核态开发环境库和用户定义开发环境库.在每一个开发环境中都分为 3 层:底层通信模块、中间适配模块和上层应用模块.由于提供多层次封装编程接口,高层接口可以最大限度屏蔽底层硬件资源的感知^[119],完善底层接口支持对系统的深度调优.

一般地,NP 进程包括 ACL(advanced C library,跨平台 C 语言库)线程、路由线程、编排调度、数据响应/请求、协议栈响应/请求以及控制响应/请求线程.HPENP 进程中,线程间通信使用自定义 msg 消息,由编排调度负责具体搬运.每个 NP 平台服务线程都有两个队列,分别是 rxq 输入队列和 txq 输出队列.所有线程的服务拥有唯一标识号用以认证该线程的身份.具体如图 6 所示.

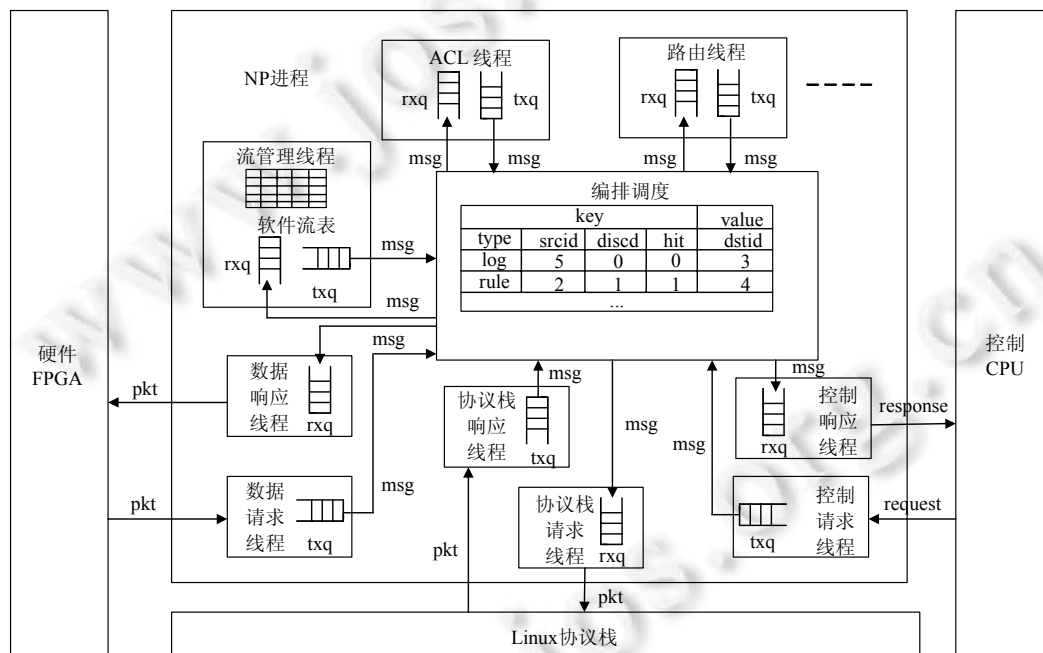


Fig.6 NP process implementation architecture

图 6 NP 进程实现架构

6.5 基于HPENP的在线智能测量及应用

基于 HPENP 架构,在线智能测量业务能够完备地满足当前智能网络管理、网络监控以及网络安全等多方面业务的基础处理需求^[120].用户可以在路由器上利用自定义功能(测量、审计、加解密等),使用 HPENP 中扩展的 API,直接在 NP 上开发相关的加速逻辑,实现基于测量的在线统计分析和机器学习功能.

在线智能测量作为各项网络业务的基础,需要满足自适应粒度的调控.在搭载下一代网络处理器的路由器

上加载自适应测量调度控制器,其根据所在网络环境中网络状态和测量任务自适应的调整测量粒度,结合主动与被动测量,对链路时延、网络带宽、丢包率、网络拓扑等进行探测以及结构化的描述.HPENP 可以加速面向多测量任务的动态调控,将测量的数据层和控制层功能进行分离,控制器通过了解各个节点的当前流量状况,全局编程调配网络测量体系结构,以进行测量任务的优化调度.通过扩展的 API 实现被动轮询和主动探测的结合,审计动态网络的视图状态.通过各项软硬件资源,建立网络环境中地址资源、流量资源表示模型,建立网络拓扑表示流量矩阵的多维动态资源视图,完成流量统计分析.通过统计分析提取网络状态特征,利用机器学习功能弹性智能的判断网络状态,完成寻路应用等场景.

6.6 HPENP原型系统构建与测试

针对上述 HPENP 的架构设计,本文将 FPGA 和 CPU 等按照设计在 PCB 板上编排,采用国产麒麟操作系统,利用标准的 19 英寸的机箱构建了 HPENP 的初步原型系统.原型系统如图 7 所示.

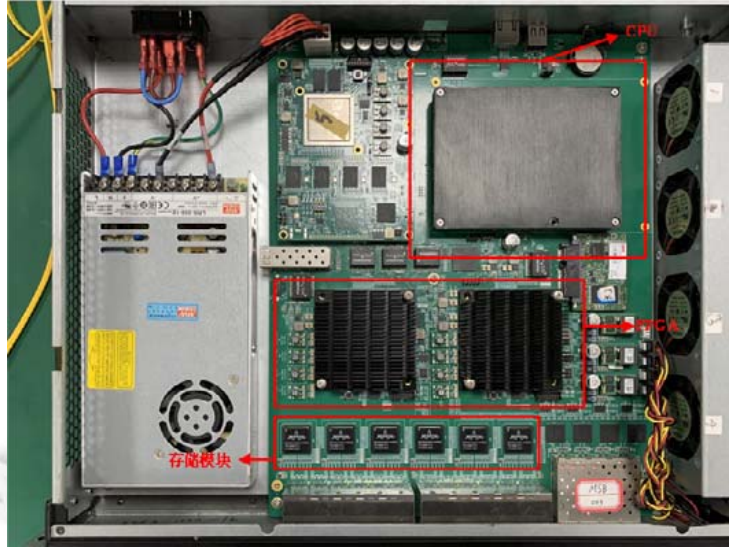


Fig.7 HPENP prototype system

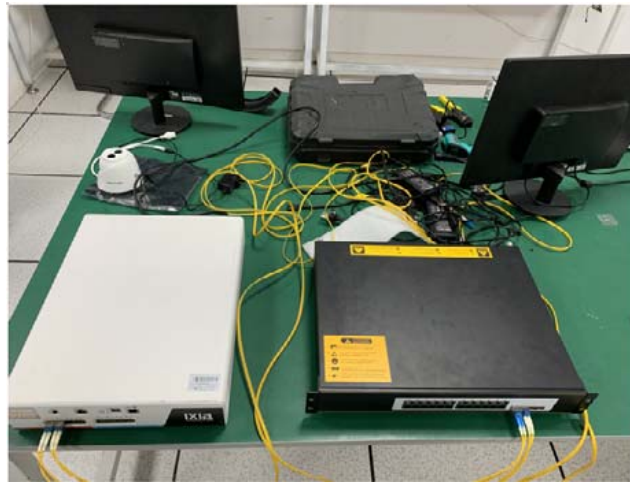
图 7 HPENP 原型系统

图 7 中,FPGA 模块主要用于流量输入解析处理和流量输出整形调度,可扩展流水线与存储模块相关联.CPU 为 4 核处理器,处理器阵列利用分组感知调度模块提高整体核间亲和性,尽量减少处理分支,保证整体性能.

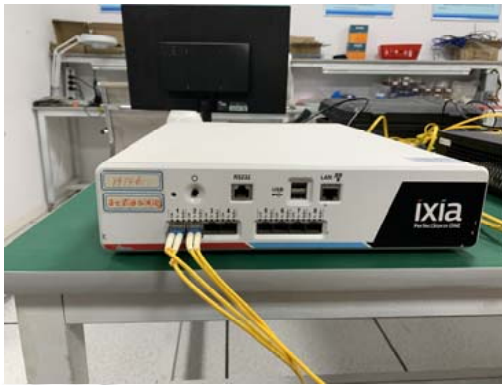
为了测试上述原型系统性能,本文利用 IXIa perfectstorm one 测试仪,按照如图 8 所示的拓扑方式搭建了实验环境,对 HPENP 原型系统相关性能进行测试.实体测试环境如下.

原型系统测试主要是网络交换基础能力测试,其包括接口指标、转发性能指标等等.针对原型系统的千兆和万兆的接口测试,本文使用了网络测试仪对千兆一套网口进行性能测试,设置使用千兆速率发送随机长度报文 1 亿个,设备开启网口回环功能后,网络测试仪的收发报文数量一致.针对原型系统的共计 10 个千兆网口用于测试交换能力,通过主机管理系统,将千兆接口的转发设置为相邻端口转发.为了验证每秒最大 bit 传输率,在网络测试仪中,将与设备相连的 2 路万兆和 2 路千兆配置了每个网口的测试报文长度为 1 518 字节,设置报文数量为 1 亿个.通过长时间测试统计,确定其交换稳定不丢包,从而可得所有网口都是线速转发.针对分组转发率指标,测试在网络测试仪中将与设备相连的 2 路万兆和 2 路千兆配置每个网口发送报文长度为 64 字节的 1 亿个报文.启动网络测试仪后,线速发送测试报文,观察并统计测试仪各网口的速率统计信息,汇总统计结果,并计算转发能力,得到原型系统拥有 30Mpacket/s 的分组转发率.针对流水线时延,利用网络测试仪的报文时间统计功能,用测试仪的一个网口发送报文,中间经过测试设备转发,然后在另一个网口回环,得到了一个报文从发送到

返回经历路径的延迟,从而设备端到端的延迟为 $28\mu\text{s}$.具体结果见表 4.



(a) 测试环境



(b) IXIa 测试仪



(c) HPENP 原型系统接口数量

Fig.8 HPENP prototype system performance test environment

图 8 HPENP 原型系统性能测试环境

Table 4 Basic performance test results of prototype system

表 4 原型系统基础性能测试结果

测试项目	测试结果
24 千兆接口自适应及万兆接口测试	接口可线速转发
交换能力	30G 交换能力
分组转发率	30Mpacket/s
转发延时	$28\mu\text{s}$

为了验证原型系统的可编程性,本文利用原型系统的用户态编程,通过定义软件模块的流表,对不可信的外部网络进行了规则约束,搭建了简易的网络安全防火墙,并建立了测试拓扑对相关功能进行验证.测试实验拓扑如图 9 所示.

在 HPENP 原型系统上,本文设置了一条包过滤的防火墙安全策略,并在内部主机 C 上启动抓包工具,配置抓取本机接收的所有报文.原型系统启动后,初始状态下,其包过滤防火墙功能默认不允许报文通过,此时,外部主机 D 构造的报文无法发送到内部主机 C,抓包工具无法抓取到外部主机 D 发送的报文;通过用户态控制器,

向原型系统下发一条源 MAC 地址为主机 D、目的 MAC 为主机 C、源 IP 为主机 D、目的 IP 为主机 C、协议为 UDP、动作为“允许”的五元组包过滤防火墙规则.继续发包后发现,外部主机 D 构造的报文成功发送到内部主机 C,抓包工具也成功抓取到外部主机 D 发送的 UDP 报文.说明原型系统被成功部署防火墙,能够按需进行相关业务的逻辑编程.

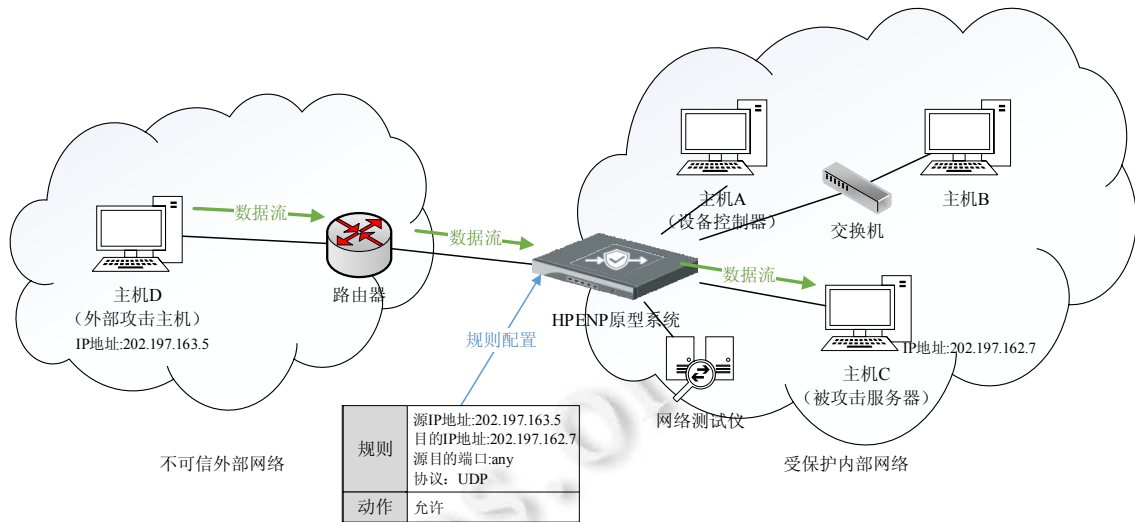


Fig.9 HPENP programmable function test

图9 HPENP 可编程性功能测试

上述结果表明:HPENP 的设计理念合理,具备高性能、可演进的优势,能够较好地兼顾灵活性和处理转发性能.相信经过进一步的研究和学习,能够完善原型系统的各项功能,升级相关架构,为下一步的处理器芯片化做好充足的准备.

7 总 结

网络处理器的设计与研究,一直是通信、网络传输领域的重点问题.硬件协处理器开发周期长、功能调整不灵活的特点,导致多样分组处理可进化需求在网络处理器中无法满足.所以,下一代网络处理器体系架构一直受到研究人员的广泛关注.针对这一背景,本文首先梳理了网络处理器的基本架构及其在当前网络环境下的发展挑战,其次介绍了利用新型可编程技术、面向新型网络体系结构、针对新型高性能业务这3个层面的业界学术界主流 NGNP 设计模式,并对不同的架构方案进行了分析和比较,指出它们的优点及不足,结合网络处理器的工业化进程,得到了相应的结论.

我们认为,下一代网络处理器研究及设计方案应当遵从以下几点意见.

- (1) 体系架构高性能可演进.当前网络由超高带宽反哺出对网络处理器的多样化分组处理需求.网络处理器体系架构设计首先考虑高性能的处理分组,进行包解析、包分类以及元数据提取存储等操作的同时完成高速转发,实现通信转发.同时,架构应当包含可动态重构硬件完成业务灵活配置,利用软硬件协同扩展的编程方式实现网络处理器的可演进性.
- (2) 分组处理可加速可优化.体系架构给予支撑,技术细节完善性能.下一代网络处理器设计应当在核间调度、多级缓存、并行处理等关键技术上有所突破,规避处理器核内大量路径分支处理开销和核间负载不均衡带来的性能损失问题.在突发流量到来时能够导流溢出流量,减少拥塞时间,高效转换地址映射,及时转发处理分组.
- (3) 硬件资源按需配置.抽象硬件资源,精准动态分配.CPU 核、FPGA 或者其他可重构逻辑资源、缓存资源应当进行统一管理,抽象成表按需、动态配置相关业务.利用业务应用到系统各类资源的重构、映

射与调度算法,提高下一代网络处理器处理性能,扩展应用场景。

- (4) 编程接口底层无关.开发环境决定使用效率.下一代网络处理器借助关系明了的开发编程接口、清晰划分的功能模块和流程规范的数据处理,完成可视、透明以及简单的业务逻辑开发与调试.软硬件通信的底层模块提供相应的 API 接口,帮助内核态分组转发设备驱动进行应用适配.向下使用底层 API 实现控制,向上丰富通信开发接口对应应用需求.
- (5) 搭载应用智能高效.高效精准的各项网络应用需要分组处理与转发的智能化.利用下一代网络处理器的硬件资源,全局编程调配各项业务.通过存储的动态多维资源视图,完成搭载应用的学习功能,实现自适应高性能的流量转发、视域测量、网络安全、QoS 保障等业务.

针对这些特点,我们提出了高性能可演进的下一代网络处理器体系结构 HPENP,介绍了软硬件协同分组处理流水线、多级缓存与分组调度、资源管理及编程接口的相关设计方案,提出了基于下一代 NP 的在线智能测量及应用,并搭建测试了 HPENP 原型系统.目前,国内外有关网络处理器的研究与设计处于快速发展阶段,虽然早期已有许多设计被产业化并大量应用,但当前也存在一些初步涉及的重要研究点.本文希望通过大量研究工作的讨论比较以及新型体系架构的设计,为下一代网络处理器的未来工作提供参考与建议.

致谢 本文的匿名评阅者对文章内容,特别是对下一代网络处理器中高性能可演进方面的完善提出了许多建设性的意见和建议,在此表示感谢.

References:

- [1] Dimitrakopoulos G, Psarras A, Seitanidis I. *Microarchitecture of Network-on-chip Routers*. New York: Springer-Verlag, 2015. 1–10.
- [2] Ogras UY, Marculescu R. *Modeling, Analysis and Optimization of Network-on-chip Communication Architectures*. New York: Springer Science and Business Media, 2013. 22–35.
- [3] *Dynamically Reconfigurable Systems: Architectures, Design Methods and Applications*. New York: Springer Science and Business Media, 2010. 1–32.
- [4] Gadre G, Badhe S, Kulkarni K. Network processor—A simplified approach for transport layer offloading on NIC. In: *Proc. of the Int'l Conf. on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2016. 21–24.
- [5] Matthews E, Shannon L. TAIGA: A new RISC-V soft-processor framework enabling high performance CPU architectural features. In: *Proc. of the 27th Int'l Conf. on Field Programmable Logic and Applications (FPL)*. IEEE, 2017. 1–4.
- [6] Jr Allen JR, Bass BM, Basso C. IBM PowerNP network processor: Hardware, software, and applications. *IBM Journal of Research and Development*, 2003,47(2.3):177–193.
- [7] Yang X, Sun Z, Li J, *et al.* FAST: Enabling fast software/hardware prototype for network experimentation. In: *Proc. of the Int'l Symp. on Quality of Service*. 2019. 1–10.
- [8] Tan ZX, Lin C, Ren FY, Zhou WJ. Analysis and research on network processor. *Ruan Jian Xue Bao/Journal of Software*, 2003, 14(2):253–267 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/14/253.htm>
- [9] Crowley P, Franklin MA, Hadimioglu H, Onufryk PZ. *Network Processor Design*. Morgan Kaufmann, 2003.
- [10] Network processor forum. 2005. <http://www.npforum.org>
- [11] Do MT, Wang H, *et al.* Sliding mode learning based congestion control for diffserv networks. *IET Control Theory and Applications*, 2016,10(11):1281–1287.
- [12] Dobbie BD, Asher DH, Kessler RE. Network processor with distributed trace buffers. U.S. Patent 9,612,934, 2017.
- [13] Konguvel E, Kannan M. A survey on FFT/IFFT processors for next generation telecommunication systems. *Journal of Circuits, Systems and Computers*, 2018,27(3):Article No.1830001.
- [14] Raj P, Raman A. Software-defined network (SDN) for network virtualization. In: *Proc. of the Software-defined Cloud Centers*. Cham: Springer-Verlag, 2018. 65–89.
- [15] Capmany J, Gasulla I, Pérez D. The programmable processor. *Nature Photonics*, 2016,10(1):6–8.
- [16] Bosshart P, Daly D, Gibb G. P4: Programming protocol-independent packet processors. *ACM SIGCOMM Computer Communication Review*, 2014,44(3):87–95.

- [17] Teubner J, Woods L. Data processing on FPGAs. In: Proc. of the Synthesis Lectures on Data Management. 2013. 8–17.
- [18] Taylor DE, Herkersdorf A, Doring A, *et al.* Robust header compression (ROHC) in next-generation network processors. *IEEE/ACM Trans. on Networking*, 2005,13(4):755–768.
- [19] Van Renterghem K, *et al.* A scalable network ASIP enabling flow awareness in Ethernet access. In: Proc. of the Int'l Conf. on Field Programmable Logic and Applications. IEEE, 2006. 649–655.
- [20] Suleiman AAZ, Khed AF, Habib SED. ASIC Implementation of Cairo University SPARC “CUSPARC” embedded processor. In: Proc. of the Int'l Conf. on Microelectronics. IEEE, 2010. 439–442.
- [21] Yoon BY, Lee BC, Lee SS. Scalable flow-based network processor for premium network services. In: Proc. of the Int'l Conf. on ICT Convergence (ICTC). IEEE, 2011. 436–440.
- [22] Zhang TR, Li XW, Wu GL, Chen JP. Design and implementation of high-speed BCP processor for OBS core node based on FPGA. In: Proc. of the 19th Annual Wireless and Optical Communications Conf. (WOCC 2010). Shanghai, 2010. 1–3.
- [23] Niu Y, *et al.* A configurable IPsec processor for high performance in-line security network processor. In: Proc. of the 7th Int'l Conf. on Computational Intelligence and Security (CIS 2011). Sanya: IEEE Computer Society, 2011. 674–678.
- [24] Veerapathap V, Nagaraja M, Kurian MZ. Network on chip design and implementation on FPGA with advanced hardware and networking functionalities. In: Proc. of the 4th Int'l Conf. on Computing. IEEE, 2014. 1–6.
- [25] Roy S. Mitigating the dark silicon phenomenon on next-generation network processor architectures. In: Proc. of the IEEE Int'l Symp. on Nanoelectronic and Information Systems (iNIS). IEEE, 2015. 124–125.
- [26] Shamani F, *et al.* FPGA implementation and integration of a reconfigurable CAN-based co-processor to the coffee risc processor. In: Proc. of the IEEE Nordic Circuits and Systems Conf. (NORCAS): NORCHIP and Int'l Symp. of System-on-chip (SoC). IEEE, 2016. 1–6.
- [27] Gadre G, Badhe S, Kulkarni K. Network processor—A simplified approach for transport layer offloading on NIC. In: Proc. of the Int'l Conf. on Advances in Computing, Communications and Informatics (ICACCI). Jaipur, 2016. 2542–2548.
- [28] Li BJ, *et al.* ClickNP: Highly flexible and high-performance network processing with reconfigurable hardware. In: Proc. of the Conf. on ACM Sigcomm Conf. ACM, 2016. 1–14.
- [29] Zhou X, Liu D, Zhai P, Liu S. A real-time fast path network processor for EtherCAT and home gateway. In: Proc. of the IEEE 4th Int'l Conf. on Integrated Circuits and Microsystems (ICICM). Beijing, 2019. 257–262.
- [30] Cao Z, Su H, Yang Q, Wen M, Zhang C. Poster abstract: A template-based framework for generating network processor in FPGA. In: Proc. of the IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS). Paris, 2019. 1057–1058.
- [31] Tanizawa K, *et al.* Ultra-compact 32×32 strictly-non-blocking Si-wire optical switch with fan-out LGA interposer. *Optics Express*, 2015,23(13):17599–17606.
- [32] Crozier K. A C-based programming language for multiprocessor network SoC architectures. In: Proc. of the Network Processor Design. 2004. 427–442.
- [33] Mu H, *et al.* P4NFV: An NFV architecture with flexible data plane reconfiguration. In: Proc. of the 14th Int'l Conf. on Network and Service Management (CNSM). IEEE, 2018. 90–98.
- [34] Cao Z, Su H, Yang Q, Shen J, Wen M, Zhang C. P4 to FPGA—A fast approach for generating efficient network processors. *IEEE Access*, 2020,8:23440–23456.
- [35] Martins RFT, *et al.* Using probabilistic data structures for monitoring of multi-tenant P4-based networks. In: Proc. of the IEEE Symp. on Computers and Communications (ISCC). IEEE, 2018. 204–207.
- [36] Paolucci F, Civerchia F, Sgambelluri A, Giorgetti A, Cugini F, Castoldi P. P4 edge node enabling stateful traffic engineering and cyber security. *IEEE/OSA Journal of Optical Communications and Networking*, 2019,11(1):A84–A95.
- [37] Yazdinejad A, Bohlooli A, Jamshidi K. P4 to SDNet: Automatic generation of an efficient protocol-independent packet parser on reconfigurable hardware. In: Proc. of the 8th Int'l Conf. on Computer and Knowledge Engineering (ICCKE). Mashhad, 2018. 159–164.
- [38] Guan B, Shen S. FlowSpy: An efficient network monitoring framework using P4 in software-defined networks. In: Proc. of the IEEE 90th Vehicular Technology Conf. (VTC2019-Fall). Honolulu, 2019. 1–5.
- [39] Jiang J, Zhang Y. An accurate congestion control mechanism in programmable network. In: Proc. of the IEEE 9th Annual Computing and Communication Workshop and Conf. (CCWC). Las Vegas, 2019. 673–677.

- [40] Benáček P, Pu V, Kubátová H. P4-to-VHDL: Automatic generation of 100 Gbps packet parsers. In: Proc. of the IEEE 24th Annual Int'l Symp. on Field-programmable Custom Computing Machines (FCCM). Washington, 2016. 148–155.
- [41] Sviridov G, Bonola M, Tulumello A, Giaccone P, Bianco A, Bianchi G. LODGE: Local decisions on global statEs in programananaable data planes. In: Proc. of the 4th IEEE Conf. on Network Softwarization and Workshops (NetSoft). Montreal, 2018. 257–261.
- [42] Pontarelli S, Bonola M, Bianchi G. Smashing SDN “built-in” actions: Programmable data plane packet manipulation in hardware. In: Proc. of the IEEE Conf. on Network Softwarization (NetSoft). Bologna, 2017. 1–9.
- [43] Azad MA, Bag S, Perera C, *et al.* Authentic caller: Self-enforcing authentication in a next-generation network. *IEEE Trans. on Industrial Informatics*, 2020,16(5):3606–3615.
- [44] Satyanarayanan M. The emergence of edge computing. *IEEE Computer*, 2017,50(1):30–39.
- [45] Ahn GJ, Gu G, Hu H, *et al.* SDN-NFV security 2019 preface. In: Proc. of the ACM Int'l Workshop on Security in Software Defined Networks and Network Function Virtualization, Co-located with CODASPY (SDN-NFV 2019). 2019. 1–4.
- [46] Brebner G. Programmable hardware for high performance SDN. In: Proc. of the Optical Fiber Communications Conf. and Exhibition. IEEE, 2015. 1–3.
- [47] Suñé M, *et al.* An OpenFlow implementation for network processors. In: Proc. of the Third European Workshop on Software Defined Networks. IEEE, 2014. 123–124.
- [48] Belter B, *et al.* Hardware abstraction layer as an SDN-enabler for non-openflow network equipment. In: Proc. of the 3rd European Workshop on Software Defined Networks. IEEE, 2014. 117–118.
- [49] Wijekoon V, *et al.* High performance flow matching architecture for OpenFlow data plane. In: Proc. of the 2nd IEEE Int'l Conf. on Network Function Virtualization and Software Defined Networks. IEEE, 2016. 186–191. [doi: 10.1109/NFV-SDN.2016.7919496]
- [50] Blaiech K, *et al.* Runtime resource allocation model over network processors. In: Proc. of the IEEE Int'l Conf. on Cloud Engineering (IC2E). IEEE, 2014. 556–561.
- [51] Seungwon S, *et al.* AVANT-GUARD: Scalable and vigilant switch flow management in software-defined networks. In: Proc. of the ACM Sigsac Conf. on Computer and Communications Security. ACM, 2013. 413–424.
- [52] Belter B, *et al.* Programmable abstraction of datapath. In: Proc. of the 3rd European Workshop on Software Defined Networks. IEEE, 2014. 7–12.
- [53] Kaljic E, Maric A, Njemcevic P. An implementation of a deeply programmable SDN switch based on a hybrid FPGA/CPU architecture. In: Proc. of the 18th Int'l Symp. on INFOTEH-JAHORINA (INFOTEH). East Sarajevo, 2019. 1–6.
- [54] Qu YR, Zhang HH, Zhou S, Prasanna VK. Optimizing many-field packet classification on FPGA, multi-core general purpose processor, and GPU. In: Proc. of the ACM/IEEE Symp. on Architectures for Networking and Communications Systems (ANCS). Oakland, 2015. 87–98.
- [55] Sun C, Bi J, Chen H, *et al.* SDPA: Toward a stateful data plane in software-defined networking. *IEEE/ACM Trans. on Networking*, 2017,25(6):3294–3308.
- [56] Chu TW, Shen CA, Wu CW. The hardware and software co-design of a configurable QoS for video streaming based on OpenFlow protocol and NetFPGA platform. *Multimedia Tools and Applications*, 2017,77(5):1–21.
- [57] Li Y, Wang G. SDN-based switch implementation on network processors. *Communications and Network*, 2013,5(3):434–437.
- [58] Sonchack J, *et al.* Enabling practical software-defined networking security applications with OFX. In: Proc. of the Network and Distributed System Security Symp. 2016.
- [59] Yazdinejad A, *et al.* P4-to-blockchain: A secure blockchain-enabled packet parser for software defined networking. *Computers and Security*, 2020,88:Article No.101629.
- [60] Mihal A, Keutzer K. A processing element and programming methodology for click elements. In: Proc. of the Workshop on Application Specific Processors. 2005. 1–13.
- [61] Nayaka RJ, Biradar RC. High performance ethernet packet processor core for next generation networks. *Int'l Journal of Next-generation Networks*, 2012,4(3):89–99.
- [62] Ihor T, Bak R. The principles of traffic processing and formation based on IXA networking processors. In: Proc. of Int'l Conf. on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET). IEEE, 2010. 218–227.
- [63] Kanada Y. Controlling network processors by using packet-processing cores. In: Proc. of the 28th Int'l Conf. on Advanced Information Networking and Applications Workshops (WAINA). IEEE, 2014. 690–695.

- [64] He P, *et al.*, Balanced locality-aware packet schedule algorithm on multicore network processor. In: Proc. of the 2nd Int'l Conf. on Future Computer and Communication. Wuhan, 2010. V3-248–V3-252.
- [65] Niu XN, *et al.* Internet traffic management based on AMCC network processor. In: Proc. of the 11th IEEE Int'l Conf. on Communication Technology (ICCT 2008). IEEE, 2008. 533–536.
- [66] Wu KH, *et al.* The research on the software architecture of network packet processing based on the many-core processors. In: Proc. of the 7th IEEE Int'l Conf. on Software Engineering and Service Science (ICSESS). IEEE, 2016. 555–559.
- [67] Tang T, Zeng X, Chen X. A multi-path reliable data transfer system based on multi-core network processors. In: Proc. of the IEEE 9th Int'l Conf. on Electronics Information and Emergency Communication (ICEIEC). Beijing, 2019. 122–125.
- [68] Blaiech K, *et al.* Data plane acceleration for virtual switching in data centers: NP-based approach. In: Proc. of the IEEE 3rd Int'l Conf. on Cloud Networking (CloudNet). IEEE, 2014. 108–113.
- [69] Bae KR, *et al.* An efficient interworking architecture of a network processor for layer 7 packet processing. In: Proc. of the Int'l Conf. on Communication and Networking. FGCN, 2011. 136–146.
- [70] Li J, Sun Z, Yan J, Yang X, Jiang Y, Quan W. DrawerPipe: A reconfigurable pipeline for network processing on FPGA-based SmartNIC. *Electronics*, 2020,9(1):Article No.59.
- [71] Li Y, Shan L, Qiao X. A parallel packet processing runtime system on multi-core network processors. In: Proc. of the 11th Int'l Symp. on Distributed Computing and Applications to Business, Engineering and Science. Guilin, 2012. 67–71.
- [72] Cascón P, *et al.* A multi-threaded network interface using network processors. In: Proc. of the 17th Euromicro Int'l Conf. on Parallel, Distributed and Network-based Processing (PDP 2009). Weimar: IEEE Computer Society, 2009. 196–200.
- [73] Sabin G, Rashti M. Security offload using the SmartNIC, a programmable 10 Gbps ethernet NIC. In: Proc. of the IEEE National Aerospace and Electronics Conf. (NAECON 2015). IEEE, 2015. 273–276.
- [74] Le Y, Chang H, Mukherjee S, Wang L, Akella A, Swift MM, Lakshman T. UNO: Uuniflying host and smart NIC offload for flexible packet processing. In: Proc. of the ACM SoCC. 2017. 506–519.
- [75] Cornevaux-Juignet F, Arzel M, Horrein P, Groléat T, Person C. Combining FPGAs and processors for high-throughput forensics IEEE CNS 17 poster. In: Proc. of the IEEE Conf. on Communications and Network Security (CNS). Las Vegas, 2017. 388–389.
- [76] Cerovic D, *et al.* Data plane offloading on a high-speed parallel processing architecture. In: Proc. of the IEEE 11th Int'l Conf. on Cloud Computing (CLOUD). IEEE, 2018. 229–236.
- [77] Huang CG, Yu X, Luo H. Research on high-speed network data stream capture based on multi-queue NIC and multi-core processor. In: Proc. of the IEEE Int'l Conf. on Information Management and Engineering. IEEE, 2010. 248–251.
- [78] Li Y, Miao R, Liu HH, Zhuang Y, Feng F, Tang L, Cao Z, Zhang M, Kelly F, Alizadeh M, Yu M. HPCC: High precision congestion control. In: Proc. of the ACM Special Interest Group Data Commun. (SIGCOMM). 2019. 44–58.
- [79] Miano S, Doriguzzi-Corin R, Risso F, Siracusa D, Sommese R. Introducing SmartNICs in server-based data plane processing: The DDoS mitigation use case. *IEEE Access*, 2019,7:107161–107170.
- [80] Choi S, Shahbaz M, Prabhakar B, Rosenblum M. λ -NIC: Interactive serverless compute on SmartNICs. In: Proc. of the ACM SIGCOMM 2019 Conf. Posters and Demos (SIGCOMM Posters and Demos 2019). New York: Association for Computing Machinery, 2019. 151–152.
- [81] Caulfield A, Costa P, Ghobadi M. Beyond SmartNICs: Towards a fully programmable cloud: Invited paper. In: Proc. of the IEEE 19th Int'l Conf. on High Performance Switching and Routing (HPSR). Bucharest, 2018. 1–6.
- [82] Firestone D, Putnam A, Mundkur S, *et al.* Azure accelerated networking: SmartNICs in the public cloud. In: Proc. of the 15th USENIX Symp. on Networked Systems Design and Implementation (NSDI 2018). 2018. 51–66.
- [83] Zhao YY, Cheng G, Li HD, Zhang WC. Active queue management algorithm for time delay demand. *Scientia Sinica Informationis*, 2019,49(10):1321–1332 (in Chinese with English abstract). [doi: 10.1360/N112019-00010]
- [84] Ficara D, Giordano S, Oppedisano F, Procissi G, Vitucci F. A cooperative PC/network-processor architecture for multi gigabit traffic analysis. In: Proc. of the 4th Int'l Telecommunication Networking Workshop on QoS in Multiservice IP Networks. Venice, 2008. 123–128.
- [85] Lu J, Wang J. Analytical performance analysis of network-processor-based application designs. In: Proc. of the 15th Int'l Conf. on Computer Communications and Networks. 2007. 33–39.
- [86] Li P, Wu X, Ran Y, Luo Y. Designing virtual network functions for 100GbE network using multicore processors. In: Proc. of the ACM/IEEE Symp. on Architectures for Networking and Communications Systems (ANCS). 2017. 49–59.

- [87] Yang X, Chen X, Jin Y. A high-speed real-time HTTP performance measurement architecture based on network processor. In: Proc. of the ICTC 2011. 2011. 744–745.
- [88] Yuan L, Chuah CN, Mohapatra P. ProgME: Towards programmable network measurement. *IEEE/ACM Trans. on Networking*, 2011,19(1):115–128.
- [89] Xie Y, Wu J, Chen J, Cui M. Dynamic measurement of task scheduling algorithm in multi-processor system. *Journal of Shanghai Jiaotong University (Science)*, 2019,24(3):372–380.
- [90] El Ferkouss O, Snaiki I, Mounaouar O, Dahmouni H, Ben Ali R, Lemieux Y, Omar C. A 100Gig network processor platform for openflow. In: Proc. of the 7th Int'l Conf. on Network and Service Management. 2011. 1–4.
- [91] Liu M, Peter S, Krishnamurthy A, *et al.* E3: Energy-efficient microservices on SmartNIC-accelerated servers. In: Proc. of the Usenix Annual Technical Conf. 2019. 363–378.
- [92] Narayan A, Cangialosi F, Goyal P, *et al.* The case for moving congestion control out of the datapath. In: Proc. of the Hot Topics in Networks. 2017. 101–107.
- [93] Avudaïammal R, Swarnalatha A, Seethalakshmi P. Network processor based high speed packet classifier for multimedia applications. *Wireless Personal Communications*, 2017,98(1):1219–1236.
- [94] Park J, Lee YS. Network processor architecture with flow-based dynamic bandwidth control for efficient QoS provisioning. *Peer-to-peer Networking and Applications*, 2015,8(4):704–715.
- [95] Li M, Ma B, Zhang WW. Research and implement of the key technology for IP QoS based on network processor. In: Proc. of the Int'l Symp. on Computer Network and Multimedia Technology. 2019. 1–4.
- [96] Saleem K, Faisal N, Zabidi MM, Mohd A. QoS provisioning for real time services for IPv6 DiffServ network using IXP-2400 Intel network processor. In: Proc. of the IEEE Int'l Conf. on Telecommunications and Malaysia Int'l Conf. on Communications, Telecommunications and Malaysia Int'l Conf. on Communications (ICT-MICC 2007). IEEE, 2007. 594–598.
- [97] Nguyen HK, Tran XT. A novel reconfigurable router for QoS guarantees in real-time NoC-based MPSoCs. *Journal of Systems Architecture*, 2019,100:Article No.101664.
- [98] Benacer I, Boyer FR, Savaria Y. Design of a low latency 40 Gb/s flow-based traffic manager using high-level synthesis. In: Proc. of the IEEE Int'l Symp. on Circuits and Systems (ISCAS). 2018. 1–5.
- [99] Iqbal MF, Holt J, Ryoo JH, de Veciana G, John LK. Dynamic core allocation and packet scheduling in multicore network processors. *IEEE Trans. on Computets*, 2016,65(12):3646–3660.
- [100] Yu Q, Znati T, Yang W, *et al.* Energy-efficient, QoS-aware packet scheduling in high-speed networks. *IEEE Journal on Selected Areas in Communications*, 2015,33(12):2789–2800.
- [101] Paul S, Pandit MK. A QoS-enhanced Smart Packet Scheduler for Multi-core Processors in Intelligent Routers Using Machine Learning. Springer-Verlag, 2018. 1–8.
- [102] Bonifaci V, Brandenburg B, D'Angelo G, Marchetti-Spaccamela A. Multiprocessor real-time scheduling with hierarchical processor affinities. In: Proc. of the 28th Euromicro Conf. on Real-time Systems (ECRTS). Toulouse, 2016. 237–247.
- [103] Jang H, Jin H. MiAMI: Multi-core aware processor affinity for TCP/IP over multiple network interfaces. In: Proc. of the 17th IEEE Symp. on High Performance Interconnects. New York, 2009. 73–82.
- [104] Ortiz A, Ortega J, Díaz AF, *et al.* Affinity-based network interfaces for efficient communication on multicore architectures. *Journal of Computer Science and Technoloty*, 2013,28:508–524.
- [105] Hanford N, Ahuja V, Farrens M, Ghosal D, Balman M, Pouyoul E, Tierney B. Improving network performance on multicore systems: Impact of core affinities on high throughput flows. *Future Generation Computer Systems*, 2016,56:277–283.
- [106] Ok SH, Moon B. A sequence-preserving packet scheduler for multi-core network processors. *Journal of Korean Institute of Information Technology*, 2019,17(2):79–85.
- [107] Roy S, Kaushik A, Agrawal R, *et al.* A high-throughput network processor architecture for latency-critical applications. *IEEE Micro*, 2020,40(1):50–56.
- [108] Satheesh A, Kumar D, Dharmalingam P, *et al.* Dynamically reconfigurable queue for Intel IXP2400 network processor. *Journal of Internet Technology*, 2017,18(1):95–101.
- [109] Rim SY, Cui ZG, Qian L. High performance packet processor architecture for network virtualization: Programmable packet processor architecture as a data flow machine. In: Proc. of the Int'l Conf. on Algorithms, Computing and Artificial Intelligence (ACAI 2018). New York: Association for Computing Machinery, 2018. 1–5.

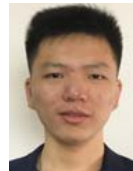
- [110] Zhou X, Liu D, Zhai P, Liu S. A real-time fast path network processor for EtherCAT and home gateway. In: Proc. of the IEEE 4th Int'l Conf. on Integrated Circuits and Microsystems (ICICM). Beijing, 2019. 257–262.
- [111] Shan Z, Zhao RC. Network processor performance evaluation model. Computer Engineering, 2007,33(22):161–163.
- [112] Parrilla L, Álvarez-Bermejo JA, Castillo E, *et al.* Elliptic curve cryptography hardware accelerator for high-performance secure servers. The Journal of Supercomputing, 2019,75(3):1107–1122.
- [113] Frost DC, Bryant SF, Filsfils C. Framework for dynamically programmed network packet processing. U.S. Patent 9,462,043, 2016.
- [114] Wienbrandt L, Kässens JC, Hübenthal M, *et al.* 1000× faster than PLINK: Combined FPGA and GPU accelerators for logistic regression-based detection of epistasis. Journal of Computational Science, 2019,30:183–193.
- [115] Svetek A, Blake M, Hermida MC, *et al.* The calorimeter trigger processor card: The next generation of high speed algorithmic data processing at CMS. Journal of Instrumentation, 2016,11(2):Article No.C02011.
- [116] Agarwal V, Patil RA, Patki AB. Architectural considerations for next generation IoT processors. IEEE Systems Journal, 2019, 13(3):2906–2917.
- [117] Pang Y, Han J, Zeng J, *et al.* Instruction set extension and hardware acceleration for SVM application toward a vector processor. In: Proc. of the Int'l SoC Design Conf. (ISOCC). IEEE, 2017. 42–43.
- [118] Aguilar J, Garcia G. An adaptive intelligent management system of advertising for social networks: A case study of facebook. IEEE Trans. on Computational Social Systems, 2017,5(1):20–32.
- [119] Meng X, Wu C, Guo M, *et al.* PAM: An efficient power-aware multilevel cache policy to reduce energy consumption of storage systems. Frontiers of Computer Science, 2019,13(4):850–863.
- [120] Enberg P, Rao A, Tarkoma S. I/O Is faster than the CPU: Let's partition resources and eliminate (Most) OS abstractions. In: Proc. of the Workshop on Hot Topics in Operating Systems. 2019. 81–87.

附中文参考文献:

- [8] 谭章熹,林闯,任丰源,周文江.网络处理器的分析与研究.软件学报,2003,14(2):253–267. <http://www.jos.org.cn/1000-9825/14/253.htm>
- [83] 赵玉宇,程光,李昊冬,张慰慈.面向时延需求的主动队列管理方法.中国科学:信息科学,2019,49(10):1321–1332. [doi: 10.1360/N112019-00010]



赵玉宇(1994—),男,博士生,CCF 学生会员,主要研究领域为 TCP/IP 协议,网络测量,网络处理器.



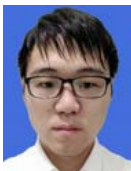
袁帅(1997—),男,硕士生,主要研究领域为 FPGA,网络测量.



程光(1973—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为网络空间安全监测和防护,网络大数据分析.



唐路(1988—),男,博士,CCF 专业会员,主要研究领域为计算机网络.



刘旭辉(1995—),男,硕士生,主要研究领域为网络处理器,命名数据网络,软件定义网络.