计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202220998
61(12): 2952−2968, 2024

ChipletNP: 基于芯粒的敏捷可定制网络处理器架构

李 韬 杨 惠 厉俊男 刘汝霖 孙志刚
(国防科技大学计算机学院 长沙 410073)
(taoli_network@163.com)

ChipletNP: Chiplet-Based Agile Customizable Network Processor Architecture

Li Tao, Yang Hui, Li Junnan, Liu Rulin, and Sun Zhigang
(*College of Computer Science and Technology*, *National University of Defense Technology*, *Changsha* 410073)

Abstract In order to meet different requirements of performance, flexibility and quality of service in new network scenarios such as 5G and 8K video, network processor (NP) becomes increasingly complex and diverse. Traditional network processors try to integrate amount of heterogeneous processing resources such as processor cores, caches and accelerators on a single SoC (system on chip) to provide highly customizable capabilities. However, with the failure of Moore's Law and Dennard's scaling law, developing one-big network processor becomes unsustainable as it faces greater challenges in R & D cycle, cost and innovation iteration. We propose a novel agile customizable architecture for network processor, namely ChipletNP, which decouples heterogeneous resources and uses Chiplet technology to quickly customize new NPs by combining existing mature chip products. ChipetNP is highly flexible as it has an agile switching network which can connect diverse heterogeneous resources with high throughput and predictable delay. We develop a network processor chip, i.e., YHHX-NP, based on ChipletNP architecture, which integrates commercial CPU, FPGA (field programmable gate array) and agile switching chip. Our results show that ChipletNP supports various emerging network functions such as SRv6 (segment routing over IPv6) with ultra-low latency (<2.82 μs), and achieves more than two times energy efficiency improvement compared with commercial chips.

Key words network processor (NP); Chiplet technology; agile switch; packet processing; heterogeneous resource

摘 要 5G, 8K 视频等新业务类型不断涌现,使得网络处理器(network processor, NP)的应用场景日趋复杂多样.为满足多样化网络应用在性能、灵活性以及服务质量保证等方面的差异化需求,传统 NP 试图在片上系统(system on chip, SoC)上集成大量处理器核、高速缓存、加速器等异质处理资源,提供面向多样化应用场景的敏捷可定制能力.然而,随着摩尔定律和登纳德缩放定律失效问题的逐渐凸显,单片 NP 芯片研制在研发周期、成本、创新迭代等方面面临巨大挑战,越来越难以为继.针对上述问题,提出新型敏捷可定制 NP 架构 ChipletNP,基于芯粒化(Chiplet)技术解耦异质资源,在充分利用成熟芯片产品及工艺的基础上,通过多个芯粒组合,满足不同应用场景下 NP 的快速定制和演化发展需求.基于 ChipletNP 设计实现了一款集成商用 CPU、FPGA(field programmable gate array)和自研敏捷交换芯粒的银河衡芯敏捷 NP 芯片(YHHX-NP).基于该芯片的应用部署与实验结果表明,ChipletNP 可支持 NP 的快速敏捷定制,能够有效承载 SRv6(segment routing over IPv6)等新型网络协议与网络功能部署.其中,核心的敏捷交换芯粒相较于同级商用芯片能效比提升 2 倍以上,延迟控制在 2.82 μs 以内,可以有效支持面向 NP 的 Chiplet 统一通信与集成.

收稿日期: 2022-12-07; **修回日期**: 2023-09-26
基金项目: 国家自然科学基金项目(62002368)
This work was supported by the National Natural Science Foundation of China (62002368).
通信作者: 杨惠(huihui19870124@126.com)

关键词 网络处理器；芯粒技术；敏捷交换；分组处理；异构资源

中图法分类号 TP393

网络处理器（network processor, NP）是网络领域专用可编程芯片，提供高性能分组转发处理能力，广泛应用于路由器、中间盒、智能网卡等网络设备完成网络通信领域的各种任务，如分组处理、协议解析、路由查找、声音/数据汇聚、防火墙、QoS（quality of service）等[1]. NP 是一种功能面向网络领域设计的集成电路. 与通用 CPU 类似，NP 通常是软件可编程的，以提供高性能分组转发处理能力，可应用于多种网络设备和系统.

在 5G、物联网、云计算等新技术发展的驱动下，新型网络应用，如虚拟现实、自动驾驶、网络直播等不断涌现. 这些应用导致互联网流量爆炸式增长，也使得 NP 的应用场景日趋复杂多样，要求 NP 芯片必须不断迭代，才能满足多样化网络应用在性能、灵活性以及服务质量保证等方面的差异化需求.

传统单片 NP 内置大量处理器核、高速缓存、加速器等异构处理资源，通常采用可配置的接口控制器，如以太网、SATA（serial advanced technology attachment）、PCIe（peripheral component interconnect express），以及复用高速串行总线口 SerDes（serializer and deserializer），以提高控制器利用率和节省资源面积，为多样化应用场景提供敏捷可定制能力. 如恩智浦的 LX2160A[2]，集成 16 个 Arm®Cortex®-A72 核以及针对 L2/L3 分组处理、50 Gbps 的安全引擎、100 Gbps 的压缩/解压缩引擎、流量管理和服务质量优化的数据通路加速单元，支持 116 Gbps L2 层以太网交换和支持 100 Gbps 速率的以太网控制器. LX2160A 的 24 个 SerDes 通道可以配置为 100 GbE/40 GbE/25 GbE/10 GbE 速率. 然而 SerDes 设计常伴有各种串扰，如噪声、抖动等问题，影响流片成功率. 相对于通用多核而言，硬件加速器部分专用性高、应用难度大，同样需要多次流片才能成熟. 这些因素造成 NP 设计复杂、对工艺要求高、流片成本高、迭代周期长，难以与设备和应用需求的演进速度保持同步，敏捷可定制能力较弱.

随着摩尔定律、登纳德缩放定律失效问题逐渐凸显，以及芯片制程演进放缓，导致芯片设计难度更高、流程更加复杂、全流程设计成本大幅度增加. 国际商务战略公司调查数据显示，22 nm 制程之后的每一代技术设计成本（包括 EDA、设计验证、IP 核、流片等）增加均超过 50%，7 nm 总设计成本约 3 亿美元，预计 3nm 工艺成本将增加 5 倍，达到 15 亿美元. 与此同时，NP 相比于一般单片集成电路，设计更加复杂，设计、验证和管理成本的增加给片上系统（system on chip, SoC）的规模带来巨大压力，研制周期通常需要 5 年或者更长时间. 例如，博通公司的 XLPNP，由于设计复杂、投片 7 次，投入近 1 亿美元. 由此可见，传统单片 NP 芯片研制在研发周期、成本、创新迭代等方面面临巨大挑战，越来越难以为继.

学术界和工业界的工作表明，通过使用先进的封装技术，即芯粒（Chiplet）技术，将多个异构芯粒在封装级紧密集成，如 CPU，FPGA（field programmable gate array），是解决摩尔定律和登纳德定律失效的一种有效途径. 通过这种方式，开发团队可以将芯片功能和技术发展路线解耦，使用最先进的技术持续交付产品. Chiplet 技术通过将一块完整芯片的复杂功能进行分解，异构集成多个模块化芯粒. 其中，这些芯粒可以由不同厂家、不同工艺的制程制造. 模块化集成方式可以有效提高芯片的研发速度、降低研发成本和芯片研制门槛，使得芯片研发聚焦于算法和核心技术，提高行业整体创新水平和能力. 脸书等公司推动的开放计算项目（open computer project, OCP）在 2018 年末积极启动了开放领域特定架构（open domain-specific architecture, ODSA）研究[3]，试图开发完整 Chiplet 体系结构的接口堆栈，创建一个 Chiplet 的开放市场. ODSA 构建兼容网络堆栈的每一层都有多个选项，以便根据所连接的芯粒类型与现有技术兼容，通过定义开放的标准化互连接口，使得 Chiplet 芯片集成的裸片可以互操作，支持不同供应商的裸片自由组合，以构建灵活的芯片系统. Chiplet 技术在芯片涉及的研发周期、成本、性能、灵活等多个维度提供可定制性和可优化性，为 NP 研制提供了一条可行的技术路线. ODSA 面向通用计算，资源种类丰富，但未考虑高性能网络处理在加速、互连、耦合处理等方面的特殊需求. 目前尚未有基于 Chiplet 的 NP 架构研究工作.

本文提出采用 Chiplet 技术构建新型敏捷可定制 NP 架构——ChipletNP，利用 Chiplet 技术将网络处理所需的异质资源解耦，可使用成熟芯片产品及工艺，通过多个芯粒组合，采用先进封装工艺快速定制 NP 芯片，满足不同场景下 NP 快速定制和演化发展需求.

首先，分析 NP 异质资源需求，抽象网络处理各类功能及常用各类硬件资源，提出基于交换、处理、

加速耦合等芯粒的 ChipletNP 体系结构. 其次, 提出实现芯粒间统一互连的敏捷交换技术, 支持多匹配动作分组处理模型以及级联扩展. 再次, 基于敏捷交换技术设计并流片实现了的敏捷交换芯粒, 相较于同级商用芯片能效比提升 2 倍以上, 延迟控制在 2.82 μs 以内, 可有效支持面向 NP 的 Chiplet 统一通信与集成. 最后, 基于 ChipletNP 架构设计实现了一款 NP 芯片——银河衡芯敏捷(YHHX-NP), 该芯片集成商用 CPU、FPGA 和自研敏捷交换芯粒, 并对其开展相关功能验证和实验评估. 实验结果表明, 基于 ChipletNP 架构可实现 NP 的敏捷定制, 能够有效协同片内异质芯粒资源实现复杂的网络处理, 具有强大的可编程能力, 能够承载 SRv6(segment routing over IPv6)等新型网络协议与网络功能的部署.

1 相关工作

本节从 NP 技术、Chiplet 技术、可编程交换芯片技术和专用硬件加速技术 4 个方面展开.

1.1 NP 技术

传统 NP 根据采用的指令集类型, 可以分为专用指令集和通用指令集 2 类多核 NP. 专用指令集多核 NP, 如思科 nPower X1[1]、迈络思 NPS-400[4]、博通 XLP900[5]、美满技术 HX4100[6], 大多采用某种裁剪后的 RISC 指令集作为基本指令, 并根据报文处理的特点拓展专用指令, 集成度高, 可获得与 ASIC 相媲美的处理性能, 但存在微码编程、调试难度大的缺陷. 通用多核处理器采用完整的 RISC 或者 CISC 指令集, 如恩智浦 T 系列[2]、凯为 CN8000 系列[7]、恩智浦 LS2088A[8]、高通 IPQ8069[9], 编译开发环境成熟、调试难度小, 但受限于通用处理器的缺陷, 存在吞吐率低、访存瓶颈和报文处理延时高且波动大的不足. Netronome 提出采用逻辑块(功能组件)敏捷构建其系列化的单片 NP, 在 2014—2018 年快速推出了 4 款不同配置的 NP. 然而, 该方式仍难以解决单片集成电路面临的问题和挑战.

为满足高性能和灵活应用场景的各项要求, 学术界[10-11]提出了各类提升灵活性、部署效率、处理性能、功耗的下一代 NP. 例如, 因特尔公司的 Crystal Forest 处理器采用多核+片外 QuickAssist 加速器[12]的协处理模型, 并将深度报文检测、加解密、解压缩等常用的分组处理功能卸载到 QuickAssist 加速器中. 现有的工作还提出针对通用多核处理器实现报文处理的优化技术, 如 DPDK(data plane development kit)[11-12], VPP(vector packet processing)[13], 能够比较有效地提升通用多核处理报文的吞吐量并降低处理延时. Li 等人[14]提出基于 FPGA 加速平台的高柔性和高性能架构——clickNP, 利用 FPGA 良好的可编程性和处理性能, 可获得 200 MPPS(million packets per second)的处理性能.

传统 NP 在单颗芯片上集成大量处理器核、加速器等部件, 设计复杂、研发周期长、成本高, 难以依据设备和应用需求推进速度同步、快速的迭代, 敏捷可定制能力较弱.

1.2 Chiplet 技术

Chiplet(又称小芯片或芯粒), 试图通过将多个可模块化芯片(主要形态为裸片(die))通过内部互连技术集成在一个封装内, 构成专用功能异构芯片, 从而解决芯片研制涉及的规模、研制成本以及周期等方面的问题[15]. 通过采用 2.5D, 3D 等高级封装技术, Chiplet 可以实现高性能多芯片片上互连, 提高芯片系统集成度, 扩展其性能、功耗优化空间. 例如, AMD 使用多达 4 个齐柏林飞艇(Zeppelin)芯片[16]构建第 1 代 EPYC 处理器(那不勒斯), 使用中央 IO 芯片来实现具有无限架构的系统级互连. 英特尔公司提出了它们的异质芯片集成计划, 可编程交换芯片 Tofino2[17]采用 Chiplet 技术, 将网络芯片高速 SerDes IO 模块与核心逻辑分离, 提供更多针对功耗优化的布局选择.

脸书等公司推动的 OCP 也在 2018 年末积极启动了 ODSA 研究, 试图开发完整体系结构的接口栈来创建一个 Chiplet 的开放市场, 通过定义开放的标准化接口, 使得 Chiplet 芯片中集成的裸片可以互操作, 以支持不同供应商的裸片自由组合, 构建更为灵活的芯片系统. ODSA 提倡通用化架构和强调兼容性, 而 NP 在延迟、带宽、处理模型方面, 对资源种类和连接方式有特殊要求. 英特尔、AMD、高通、ARM、三星、台积电、日月光等大厂, 以及谷歌云、微软等公司于 2022 年 3 月 2 日宣布了一项新技术标准 UCIe(universal Chiplet interconnect express). 但其仅针对物理层通信制定的标准, 且与工艺、功耗以及性能紧密相关, 尚未提出链路层及以上接口及协议, 依赖沿用并扩展已有如 PCIe 标准接口与协议. 使用特定领域加速器和芯片的好处包括减少成本和上市时间、减少电路板面积和功耗, 且加速器可以用于组装集成多种系统.

近几年演化出的 Chiplet 网络处理芯片、英特尔(原 Barefoot)Tofino 2(12.8 Tbps)交换芯片采用交换逻辑芯片与高速 SerDes 接口模块芯片组合的 Chiplet 方式实现. 但其仅基于 Chiplet 技术将网络芯片高速 SerDes IO 模块与核心逻辑分离, 未充分解耦资源, 尚

未形成 Chiplet 架构.

Chiplet 为 NP 设计提供了新的解决思路. 相较于芯粒的通用处理器设计, 在 NP 中应用 Chiplet 技术时, 需要考虑到网络应用处理延时、吞吐率的差异化需求, 同时需设计相关处理模型简化网络应用的开发和部署.

1.3 可编程交换芯片技术

与传统固定功能的交换芯片不同, 可编程交换芯片将报文处理抽象为匹配和动作 2 种操作. 其中, 匹配可根据用户需求配置待匹配的关键字, 如源/目的 IP 地址、五元组信息等, 以及待查找的规则内容; 动作则是根据匹配结果执行报文处理. 例如, 博通的 BCM56000 系列以及美满技术的 Teralynx 系列, 将报文交换拆分成固定处理逻辑(如 L2 层交换)、ACL 过滤, 以及自定义处理逻辑. 并在自定义处理逻辑中设置 3 态内容寻址存储器(ternary content addressable memory, TCAM)以支持模糊匹配, 以及通过 PCIe 连接 ARM A72 处理器支持复杂处理动作.

为更大限度地提升传统 NP 的处理性能且保持良好的灵活性, 协议无关的可编程交换芯片应运而生. 可编程交换模型, 如可重构匹配表(reconfigurable match table, RMT)[17]、分解的可重构匹配表(disaggregated reconfigurable match table, dRMT)[18], 其思想是将报文处理抽象为“匹配-动作表”(match-action table, MAT), 匹配阶段输入的关键字是从报文头提取的任意字段, 而执行的处理动作由命中的规则决定. 允许在不需要修改硬件处理逻辑条件下, 通过配置匹配-动作规则表, 实现用户定制的报文处理功能.

与博通、美满技术等公司的可编程交换芯片不同, Barefoot Tofino 交换芯片包含可编程解析器, 识别报文类型并提取关键字组成元数据(metadata), 并设置由执行超长字执行单元组成的动作处理部件, 可以组合支持大量复杂的报文处理功能.

但由于可编程交换芯片针对 L2 层交换(或 Openflow 协议), 报文处理逻辑相对单一, 仅采用匹配-动作处理抽象, 难以支持多样化网络应用需求, 例如, TCP 重放攻击检测需要维护 TCP 连接、序列号等状态信息.

因此, 现有商用网络交换芯片面向数据中心、园区网、企业网等大规模固定网络交换需求, 支持大规模、高性能的网络交换, 支持百万级流量的转发、交换管理, 难以满足 ChipletNP 高能效、低功耗的集成需求.

现有的商用可编程交换芯片往往采用 RMT 多级匹配-动作架构, 强调动作的丰富可编程性, 但其匹配部分能力较弱, 不支持超宽匹配, 不能满足流量精细控制以及能效需求. 与可编程交换芯片的应用场景不同, NP 更强调软件的可编程性, 分组处理动作大多部署在耦合加速芯粒或 CPU 芯粒上处理, 因此需要强调匹配部分的能力, 弱化动作的丰富性, 因此可编程交换芯片也难以作为敏捷交换芯粒实现最优选择.

1.4 网络加速技术

为提升网络处理吞吐率和降低通信延时, 学术界和产业界提出了专用网络处理加速器/芯片. 例如, 微软公司将基于 FPGA 的智能网卡(AccelNet)[19]应用在数据中心, 可将端到端通信延时控制在微秒级. 英伟达公司网络的 CX5 系列智能网卡[20], 支持 ASAP2 加速, 可以把网络相关工作(快速分组处理路径)卸载到 eSwitch, 而慢速分组处理路径及控制面仍然由主机端 CPU 处理. 数据中心也使用一些基于 NP 的智能网卡(如美满技术公司的 LiquidIO II[21]系列、华为公司的 IN300[22]系列), 用于加速原先在通用多核上部署的网络功能.

但由于传统智能网卡将消耗大量宝贵的 CPU 内核来进行流量的分类、跟踪和控制. 这些昂贵的 CPU 内核是为通用应用程序而设计的, 而并非为了网络数据包的查找和管理. 英伟达公司提出新一代智能网卡 DPU(data processing unit)[23], 将更多原先在软件中实现的处理功能, 如协议栈, 卸载到网卡上实现, 并可以旁路 CPU 直接将数据送给 GPU, 从而实现不同 GPU 节点间低延时数据传输, 使多节点并行计算效率更高.

因特尔公司也提出了自己的下一代智能网卡 IPU(infrastructure processing unit)[24]. 与 DPU 着重关注数据处理不同, IPU 更偏向于将用户业务负载和云厂商基础负载分离, 将基础负载的一些工作迁移到 IPU 上实现, 并发布了基于因特尔公司的 Agilex FPGA 和 Xeon-D 片上处理器的 IPU, 即 Oak Springs Canyon[25], 以及基于 ASIC 第 2 款 IPU, 即 Mount Evans.

DPU、智能网卡、IPU 用于卸载端侧网络应用, 对灵活性的需求相对较强以适应不同类型的应用需求, 但对处理能力的需求相对较弱[26]. 同时, 相关技术还未采用 Chiplet 技术, 集成大量处理单元带来极大的开发难度和成本, 且往往采用单一或者少数集中处理资源, 难以兼顾各类网络应用处理需求.

2 ChipletNP 架构与处理模型

Chiplet 技术通过将多个模块化的异构芯粒进行

组合，采用先进的多芯片封装工艺形成一块具有完整功能的芯片. 这些芯粒可由不同厂家、不同工艺的制程制造，模块化集成方式可以有效提高芯片的研发速度、降低研发成本和芯片研制门槛，使得芯片研发聚焦于算法和核心技术，提高行业整体创新水平和能力.

虽然 ODSA 已提出通用可供参考的芯粒资源集成方案，但考虑到 NP 芯片有其特殊的领域特性和资源需求特性，主要面向网络处理领域. 因此，在 NP 中应用 Chiplet 技术时，不仅需要考虑带宽，更需要侧重考虑网络处理资源连接对于延迟、堆叠、统一互连的要求，以支撑网络应用的精细部署. 芯粒化的 NP 相较于普通异质芯粒资源互连侧重考虑高带宽，NP 的芯粒间互连基于专用敏捷交换芯粒，作为 NP 的互连核心，实现高带宽、低延迟，支持网络处理低延时和分组流量的精细部署，实现各异质芯粒之间分组和中间处理结果的统一互连与通信，同时承载网络处理的快速路径实现交换堆叠，是 NP 获得极高处理性能的关键.

另一方面，基于异质芯粒构建的 NP，需支持高效的分组处理模型，才能实现异质资源的一体化处理和有效部署网络功能.

本文提出新型敏捷可定制 NP 架构 ChipletNP，基于 Chiplet 技术解耦异质资源，分析 NP 的异质资源组成，充分利用成熟芯片产品及工艺，设计专用敏捷交换芯粒实现多芯粒互连. 同时，提出基于 ChipletNP 的多匹配动作增强模型，满足不同应用场景下 NP 的快速定制和演化发展需求.

2.1 NP 资源集成

首先针对传统单片 NP 以及工业界和学术界的 NP 集成异质资源展开分析，为 ChipletNP 组成架构设计提供支撑.

2.1.1 传统单片 NP 集成资源

商用高性能 NP 架构，以典型芯片 LX2160A 为例，如图 1 所示，集成了 16 个运行频率达 2.2 GHz 的 Arm Cortex-A72 CPU 内核，8 MB 2 级缓存，共享 8 MB 片上缓存；集成支持 130 Gbps L2 层以太网交换模块，最多 16 个以太网端口；复用的 24 个 SerDes 通道可以配置为 100 GbE/40 GbE/25 GbE/10 GbE 端口；集成加速引擎模块，包括 50 Gbps 安全加速器、100 Gbps 数据压缩/解压缩引擎；集成多种外设接口包括 SD, eMMC 等；集成支持 ECC 的 2 个 72 b DDR4 可通过通用多核间的报文级并行、通用多核内部指令级并行、通用多核和以太网交换单元与加速引擎模块的任务级并行以保证高性能，面向 L2, L3 层处理，可根据应用场景处理性能需求，提供 12 核(LX2120A)和 8 核(LX2080A)版本.

由此可见，传统单片 NP 通常集成通用多核处理模块、以太网交换模块、可配置高速接口、加速引擎 4 大部分，通过内部系统高性能总线互连，新型网络协议和功能只能部署到通用多核上执行，处理性能和支撑能力较弱.

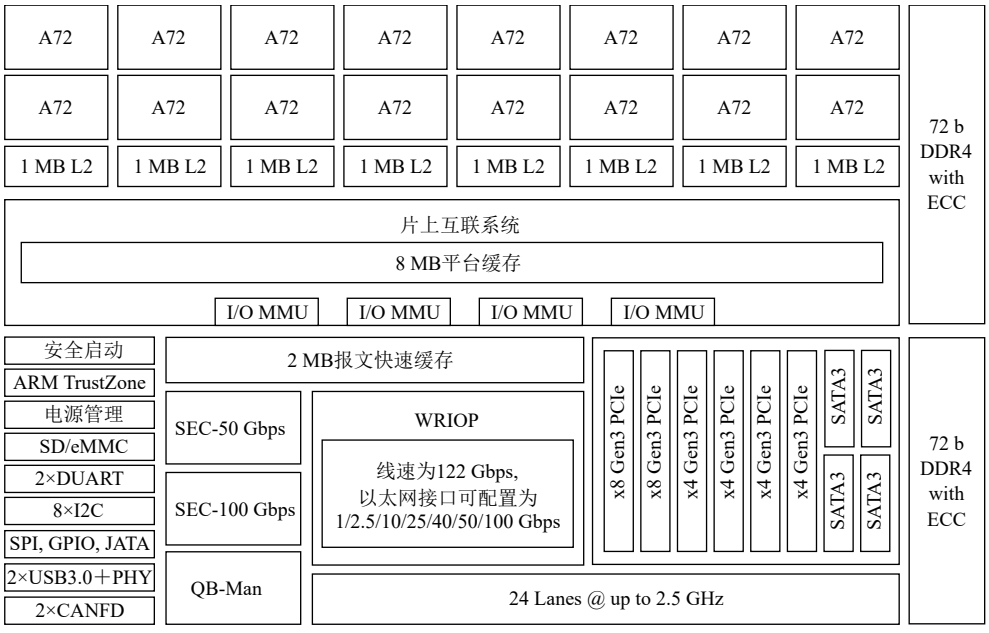


Fig. 1 LX2160A architecture

图 1 LX2160A 体系架构

2.1.2 可编程网络处理平台集成资源

现有可编程网络处理平台存在多种异质资源集成形式，表 1 给出这些平台的特性对比.

Table 1 Features Comparison of Network Processing Platform Architechture

表 1 网络处理平台体系结构的特征对比

实现架构	吞吐率	延迟	可扩展性	编程语言	OS	功耗
多核 CPU（+ASIC）	低	高	好	C/C++等	支持	高
专用 CPU（+ASIC）	高	高	好	汇编/C	支持	高
可编程 ASIC	高	低	较差	P4	不支持	低
FPGA（+CPU）	较高	低	好	Verilog/VHDL 等	部分支持	较低

1)多核 CPU(+ASIC)加速器. 典型代表有凯为 CN8000, 该加速器通过集成 ASIC 专用硬件加速器或扩展专用分组处理指令, 能够极大地提升报文处理性能, 但由于 ASIC 具有繁杂的设计、开发、调试流程, 难以满足灵活部署新网络功能或者加速各类网络应用的需求. 随着摩尔定律的失效, 通过堆积 CPU 核数扩展性能的方式已难以满足网络流量的增长速率, 需要采用新的分组处理架构.

2)专用 CPU(+ASIC). 典型代表有迈络思 NPS400. 专用 CPU 能够大大提升吞吐率, 然而专用 CPU 编程能力欠佳, 往往采用基于汇编指令的微码编程, 依旧存在处理延迟高的通病.

3)可编程 ASIC. 典型代表有 RMT, 其基于动态可配置流水线模型, 不需要修改硬件处理逻辑. 通过配置匹配-动作规则表, 可灵活定制报文处理功能. 但其仍存在 2 点不足: 一是流水线均匀分配资源不符合实际需求, 存在资源浪费; 二是针对 L4 层及 L4 层以下的无状态分组处理功能, 无法支持复杂的有状态处理.

4)FPGA(+CPU). 典型代表有 SwitchBlade[27], 动态可重构流水线可支持在线重组基本硬件模块来构造所需的分组处理功能. 这种架构虽然能够提升灵活性以及避免综合硬件逻辑的时间开销, 但仍存在 2 点不足: 一是通用基本硬件模块的动作类型较少, 导致可以组合获得的处理动作有限; 二是不同的网络功能所需的分组处理模块可能分布不均, 如果同时部署这些网络功能, 则存在空闲模块造成资源浪费而公共模块出现性能瓶颈的问题.

综上所述, 上述处理平台及资源通常面向具体网络应用领域构建, 缺乏统一的互连架构以及处理模型, ChipletNP 设计考虑并借鉴上述异质资源集成特点, 并提供统一的互连架构和分组处理模型.

2.2 ChipletNP 体系架构

综合考虑 NP 在性能、可编程性、可扩展性以及 QoS 等方面的需求, ChipletNP 集成异质芯粒, 以协同完成网络分组处理业务, 如图 2 所示.

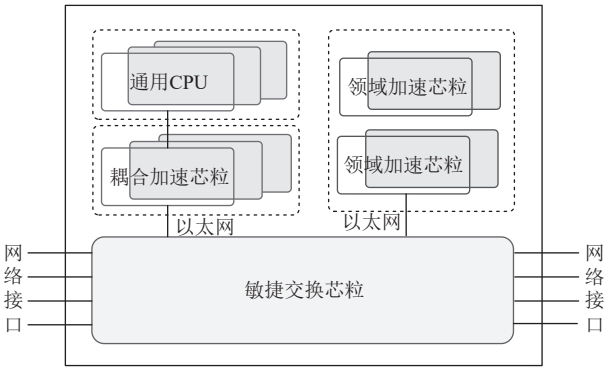


Fig. 2 ChipletNP architecture

图 2 ChipletNP 体系架构

1)敏捷交换芯粒

Chiplet 系统集成的核心——敏捷交换芯粒支持 NP 实现高密度和高带宽线速处理能力, 完成网络功能的加速和卸载, 实现异构芯粒资源互连和承载芯粒内交换网络, 将分组流量优化部署到异构芯粒相应的处理资源上以提供芯粒异构资源的统一连接. 同时实现网络功能的快速路径, 具有多个 SerDes 端口支持交换堆叠, 是 NP 获得极高处理性能的关键.

现有的交换芯片, 仅作为网络处理的快速路径以加速芯片, 并未针对芯粒间的互连进行优化设计, 不利于芯粒集成. 因此, 需基于交换芯片实现网络加速处理的同时, 提供统一的互连接口承载芯粒内敏捷交换网络, 才能为实现芯粒内异构资源负载均衡和芯粒资源的 Chiplet 系统集成提供基础支撑.

2)耦合加速芯粒

基于 DPU, NetDAM(network directly attached memory), FPGA 等实现网络功能紧耦合与加速处理. 耦合加速芯粒承担如网络切片、SRv6 等新业务的升级、应用加速的卸载、流量负载均衡, 实现网络编程, 弥补可配置交换芯片 ASIC 灵活性差、无法根据用户需求重构处理逻辑, 以及通用多核处理器芯片 CPU 处理性能低的不足.

3)通用 CPU 芯粒

通用多核处理器阵列芯粒支持 NP 的灵活可编程性以实现深度处理. 通用多核处理器芯粒基于商用货架可独立演化, 一般采用以太网和 PCIe 用作芯片间接口.

通用多核处理器既可以配置成流水线(pipeline)处理模式,即每个核实现一种网络功能,不同核之间按照功能服务链编排;也可以配置成RTC(run to complete)模式,即每个核均独自运行所有网络功能,无需核间的数据交互.此外,CPU阵列可用于下发报文处理规则到耦合加速芯粒或可配置交换芯粒,避免后续可以在交换芯粒或者加速芯粒中处理的报文继续送给CPU整列处理,以提升处理性能.

4)领域加速芯粒

用于实现领域特定加速逻辑,包括分组处理的定制加速器(正则表达式、加解密等)以及定制处理器(低延迟处理器(NanoPU))等,可面向领域应用进行Chiplet集成.

上述芯粒可根据业务应用需求,通过SIP(system in package), MCM(multi-chip module)等多芯片封装技术封装在公共基板上,形成系统级NP解决方案.

ChipletNP基于Chiplet可组合思想,融合处理、加速、互连等异质资源,通过标准化网络接口互连通信,协同完成分组处理业务.一方面,可以有效支持资源灵活配置需求,根据不同应用领域对性能、功耗的差异化需求配置资源类型和数量,以快速定制NP芯片;另一方面,支持商用成熟资源的集成,允许各资源独立演化、升级,提供更大的体系结构设计空间.

值得注意的是,敏捷交换芯粒作为ChipletNP的核心互连芯粒,需要支持芯粒间交互分组数据和中间处理结果,实现各处理平面之间的统一互连与通信.

2.3 ChipletNP分组处理模型

ChipletNP的分组处理采用多匹配动作(multiple matches & action, MMA)增强模型(multiple matches & action+, MMA+),综合了可编程交换芯片RMT流水处理模型与传统NP的RTC模型的优点.

如图3所示,MMA模型映射到敏捷交换芯片上,有状态的增强处理(action+, A+)可映射到耦合交换芯粒和CPU芯粒上.MMA+模型将分组处理数据平面划分为3个子平面,包括快速交换、在线加速以及深度处理,并使用DMID(destination module identifier)实现软硬件模块间的数据交互.

1)快速交换子平面

快速交换子平面基于敏捷交换芯粒实现,该平面由一系列被抽象为协议无关的流水阶段组成,采用协议无关的MMA抽象能够实现分类、转发、调度等各类无状态报文处理功能,具有高吞吐和低延迟的特点.多级匹配阶段实现关键字匹配功能,支持精确匹配和掩码匹配2种类型;动作阶段则是根据匹配结果执行分组处理.基于敏捷交换芯粒多级查表和多表冗余的流水线架构,能够实现细粒度流分类和查表.所有报文均先经过敏捷交换芯粒匹配查询,然后根据匹配结果决定是否需要上送给在线加速子平面或深度处理子平面处理.从而将不同类型内容的流量精细控制和部署到芯粒内异构资源上,实现更多的有状态增强处理,为芯粒内异构资源负载均衡等提供基础支撑.此外,敏捷交换芯粒流水线头部是敏捷交换可编程解析器,其识别报文类型并提取关键字组成元数据.流水线的尾部是敏捷交换协议(agile switch protocol, ASP)处理模块,可将查表结果元数据随分组一同输出.

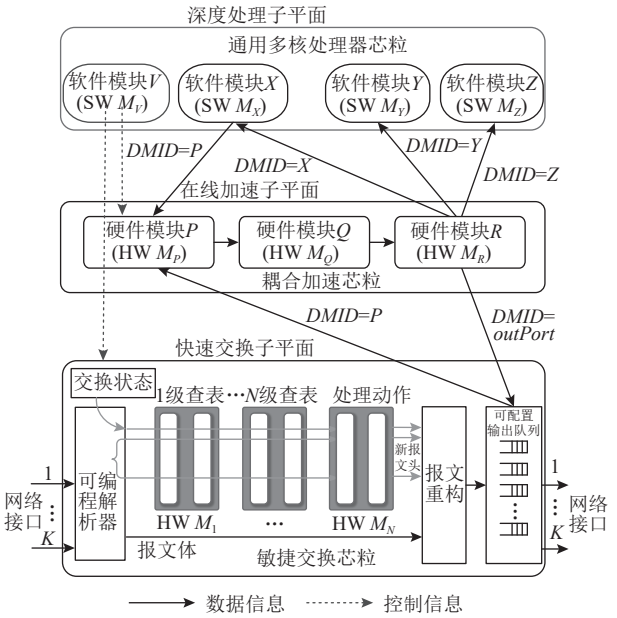


Fig. 3 Packet processing model of ChipletNP MMA+

图3 ChipletNP MMA+分组处理模型

2)在线加速子平面

在线加速子平面基于耦合加速芯粒实现,该平面利用其可编程或可重构特性来加速新业务、新协议等专用报文处理,兼备良好的处理性能和灵活性.实现2部分功能:一是与芯粒间通信相关的处理逻辑,屏蔽底层连接实现,完成与其他异构处理资源的数据、控制消息交互;二是支持快速部署网络功能的重构和支持网络功能服务链的编排,为用户提供与底层硬件无关的信号接口,允许开发者插入、删除或替换应用相关处理逻辑以快速重构新的网络功能和支持各类新协议,加速各种定制报文处理功能.

3)深度处理子平面

深度处理子平面基于通用多核处理器芯粒实现,该平面面向多样化应用场景,集成通用多核处理器

(可按需配置 CUP 核数)及大容量的存储资源，用于实现应用层协议识别、深度报文处理功能，具有极高的灵活性.

MMA+模型实现的 3 个分组处理数据子平面，可配合实现 RMT 流水处理以及 RTC 两种处理模式. 若采用 RMT 流水处理模式，这 3 个子平面紧耦合的映射执行分组处理的部分功能，通过定义标准的数据、控制消息，每个报文附加额外元数据用于 3 个处理平面的分组处理结果，以实现数据交互和共享. 若采用 RTC 处理模式，3 个子平面可解耦独立处理各自映射的流量. 因此，MMA+模型可以支持同时部署 2 种处理模式来开发的网络功能，支持 3 类资源的灵活组合配置.

3 ChipletNP 敏捷交换技术

ChipletNP 敏捷交换技术在不同应用场景和数据传输带宽、延迟、能效等方面具有不同需求，要求 ChipletNP 敏捷交换芯粒能够实现统一、可扩展和高能效的互连，支持芯粒资源的按需配置与集成，从而实现面向应用敏捷定制 NP.

具体地，敏捷交换芯粒应具有 5 种能力：1)具有芯粒间统一、标准数据交换，且能够广泛兼容连接各类异质商用芯粒能力；2)具有面向应用支持细粒度的芯粒间分组转发控制能力，将分组流量精细优化部署到芯粒相应的处理资源上；3)具有芯粒分组转发卸载加速能力，有效减轻通用多核阵列负载，实现分组快速转发；4)具有高能效、低功耗交换能力，以满足 ChipletNP 在设计空间、散热等方面的要求；5)具有堆叠扩展能力，提供异质芯粒资源可扩展能力.

如图 4 所示，敏捷交换芯粒采用标准以太网交换协议. 这是因为以太网是目前应用最为广泛的交换协议，其性能高、成本低，且 CPU 等各类处理、加速等芯片广泛集成有以太网接口.

现有商用网络交换芯片，如博通、盛科等，面向

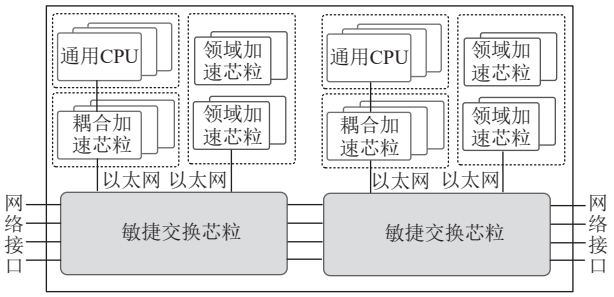


Fig. 4 Stack expansion of heterogeneous Chiplet resources based on agile switching Chiplet

图 4 基于敏捷交换芯粒的异质芯粒资源堆叠扩展

数据中心、园区网、企业网等大规模固定网络交换需求，出于市场与成本考虑，更强调实现丰富的功能协议集合，支持大规模、高性能的网络交换，支持百万级流量的转发、交换管理，难以满足上述高能效、低功耗等敏捷交换需求. 商用可编程交换芯片通常采用 RMT 多级匹配-动作架构提升芯片对分组处理的可编程灵活性. 由于该芯片更强调动作的丰富可编程性，匹配能力相对较弱，难以满足敏捷交换细粒度流量控制的需求.

与可编程交换芯片的应用场景不同，NP 更强调软件可编程性，依据 MMA+模型，ChipletNP 分组处理动作大多部署在耦合加速芯粒或 CPU 芯粒上处理. 此外，考虑到流量精细控制以及能效需求，可编程交换芯片也难以作为敏捷交换芯粒实现最优选择.

敏捷交换芯粒的实现采用 MMA 架构，如图 5 所示，主要部件包括可配置高速接口模块、敏捷交换可编程解析与封装模块、协议无关匹配-动作流水线 3 部分.

1)可配置高速接口模块. 芯片内置可配置高速接口模块，采用内置 SerDes 通道的集成化设计，基于灵活的端口配置模式、复用的 SerDes 通道来灵活支持多种速率的以太网端口，更好地支撑敏捷交换间的堆叠扩展，从而提供多样化芯粒资源集成能力.

2)敏捷交换可编程解析与封装模块. 该模块为

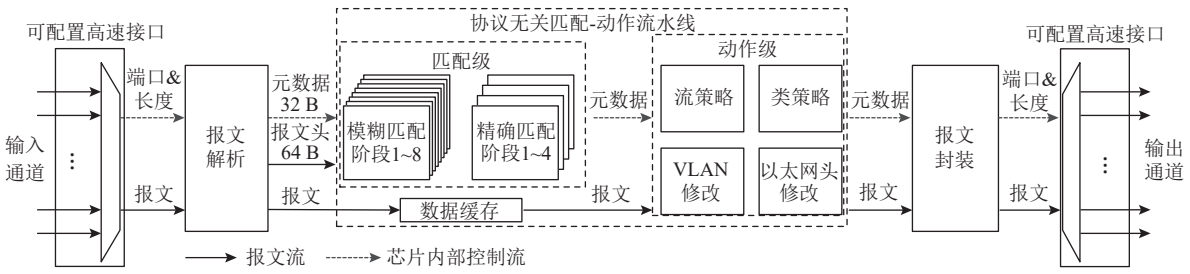


Fig. 5 Agile switching Chiplet architecture

图 5 敏捷交换芯粒架构

ChipletNP 架构提供了芯粒间的转发平面协同处理能力. 前级芯粒可通过元数据随分组一同输出的方式将内部的查表结果随原始报文分组一同输出, 后级芯粒解析该报文即可获取前级查表信息, 由此通过芯粒级联即可有效扩展报文分组处理流水线的长度和复杂度等功能处理能力, 充分发挥异构资源芯粒互连优势. 敏捷交换可编程解析能够识别标准以太网报文或带有元数据的报文, 识别报文类型并提取关键字组成元数据, 与分组一同进入协议无关匹配-动作流水线中. 封装模块可灵活配置输出报文为标准以太网报文或带有元数据的报文, 广泛兼容各类商用芯粒.

3)协议无关匹配-动作流水线. 该部分采用超宽字节掩码匹配模型提供兼容 OpenFlow 模型的可编程转发平面处理能力, 通过配置可支持灵活的匹配粒度调节, 多级匹配表提供了丰富的正交表项组合. 64 B 报文+32 B 查表信息, 能够广泛支持当前 L2~L7 各种已有协议及用户自定义协议帧的转发, 实现面向应用的细粒度分组转发控制能力, 从而支持流量的精细化部署. 支持软件定义交换, 通过配置流表可构建不同堆叠拓扑. 基于 SBV(stride bit vector)算法代替 TCAM 实现的掩码匹配可支持并行全流水. 除基本转发动作外, 协议无关匹配-动作流水线还集成了 QoS 流限速、L2 层报文头(目的 MAC, 源 MAC, VLAN)修改等功能, 以减少报文分组的处理路径延时和对其他芯粒资源的占用, 提供芯粒分组转发卸载加速能力.

4 银河衡芯敏捷网络处理器(YHHX-NP)

针对新型网络技术部署等应用场景, 基于 ChipletNP 架构敏捷定制了 YHHX-NP. 新型网络技术部署更强调 NP 的可编程性和可定制性, 意图在性能、灵活性、功耗等多方面实现最优的定制设计.

如图 6(a)所示, YHHX-NP 采用全国产基板设计和封装工艺实现, 基于多芯片封装技术集成整合通用 CPU 芯粒(某国产通用多核 CPU)、耦合加速 FPGA 芯粒(复旦微 JFM7K325T)以及敏捷交换芯粒(自研 HX-DS160 交换芯片 28 nm 工艺). 值得注意的是, 由于基于 ChipletNP 敏捷可定制架构, YHHX-NP 研制周期仅为 6 个月, 远少于单芯片 NP 芯片研制时间.

1)敏捷交换芯粒

其中自研敏捷交换芯粒 HX-DS160, 如图 6(b)所示, 是一款面向 ChipletNP 集成设计的高带宽、低功耗敏捷交换芯片,该芯片也可作为高效能以太网交换

(a) YHHX-NP原型芯片 (b) HX-DS160原型芯片

Fig. 6 Network processor chips designed based on ChipletNP

图 6 基于 ChipletNP 设计的网络处理器芯片

以及智能网络接口芯片独立应用, 独立应用封装尺寸为 27 mm×27 mm 的 BGA676. 芯片采用自主设计方式, 依托 28 nm 工艺节点流片, 具有 160 Gbps 交换性能(单向), 提供 16 个万兆以太网接口, 外部接口可配置为 2×40 Gbps+8×10 Gbps+8×1Gbps; 支持全端口线速转发交换处理, 典型功耗仅 8.5 W; 支持堆叠扩展模式. 与应用广泛的某国产 120 Gbps 以太网交换芯片相比, 性能功耗比提升 2 倍以上(从 7.5 Gbps/W 提升至 18.8 Gbps/W).

HX-DS160 采用可编程的协议无关 MMA 交换架构, 以及采用超宽比特搜索匹配引擎, 支持宽度 768 b 的关键字搜索. HX-DS160 可通过 40 Gbps/10 Gbps/1 Gbps 数据接口进行带内配置, 也支持 SPI(serial peripheral interface)和 IIC(inter-integrated circuit)接口进行轻量化管理配置, 提供无管理 CPU 的集成方式以满足敏捷定制需求.

2)耦合加速芯粒

ChipletNP 耦合加速芯粒基于国产高性能 FPGA(复旦微 JFM7K325T)实现. 为提升网络处理吞吐率和降低通信延时, 常用的耦合加速芯粒资源有 DPU, IPU, FPGA 等, 相较于 DPU 与 IPU, ChipletNP 的优势体现在 2 方面: 一是 DPU 与 IPU 受限于网络设备应用场景, 这两者用于卸载端侧网络应用, 适应不同类型的应用需求, 处理性能相对较弱; 二是用户定制能力方面, FPGA 具有良好的可重构性, 有利于提高整个 NP 的敏捷定制能力. 因此基于高性能 FPGA 实现的 ChipletNP 耦合加速芯粒, 能够兼备良好的处理性能和灵活性.

ChipletNP 耦合加速芯粒实现 2 部分功能: 一是与芯粒间通信相关的处理逻辑 FPGA OS, 屏蔽底层连接实现, 完成 FPGA-CPU 数据、控制消息交互功能. 交互部件可以基于 PCIe 总线, FPGA 向上提供数

据收发 API(application interface)，实现 FPGA 与 CPU 交换报文和元数据，是 FPGA-CPU 协同处理的基础. 由于与芯粒间耦合协同优化设计相关性较小，此处不做详述. 二是实现可重构分组流水线，这也是耦合加速芯粒体现耦合性的重要部分，是支持 ChipletNP MMA+分组处理模型的关键. 采用模块化设计框架，基于可编程模块索引机制，可将报文处理流水线抽象为一系列具有相同接口的功能模块，实现网络功能服务链的编排以构建所需的网络功能. 同时，为 NP 用户提供一组与底层硬件无关的信号接口，允许开发者插入、删除或替换应用相关处理逻辑以快速重构新的网络功能，从而支持各类新协议实现加速各种定制报文处理功能，可重构分组处理流水线.

基于 FPGA 实现的可重构分组处理流水线的支持调度包含 FPGA 硬件功能模块、敏捷交换芯粒硬件功能模块和通用 CPU 的软件进程 3 部分. 其中硬件功能模块采用串行组织，即图 3 中的 M_1, M_2, …, M_N, M_P, M_Q, M_R. 软件进程则运行软件功能模块，即图 3 中的 M_V, M_P, M_Q, M_R 并支持动态加载和移除，其中 M_V 为配置模块.

可重构分组处理流水线处理报文的流程为 4 步：①网络端口接收报文，并送给敏捷交换硬件流水线的第 1 个模块 M_1，然后再依次经过后续的匹配模块进行匹配，查表匹配结果随分组一同输出，即为每个报文附加额外的元数据，方便模块间共享中间处理结果，元数据中携带 *DMID*，用于指示处理当前报文的下一个模块. 根据查表匹配结果，将流量送达基于耦合加速芯粒实现的可重构分组处理流水线中完成硬件部分的处理. ②基于耦合加速芯粒实现的可重构分组处理流水线会根据 *DMID* 的数值判断是将报文直接从网络接口输出还是送软件模块 M_X(即 *DMID*=*x*)或 M_Y, M_Z 进行深度处理. ③每个软件模块在完成报文处理后，均会将报文继续送回基于耦合加速芯粒实现的可重构分组处理流水线中. ④报文可再次经过所有硬件模块处理，或者根据 *DMID* 直接跳到敏捷交换芯粒硬件最后一个功能模块 M_N，然后根据输出端口转发.

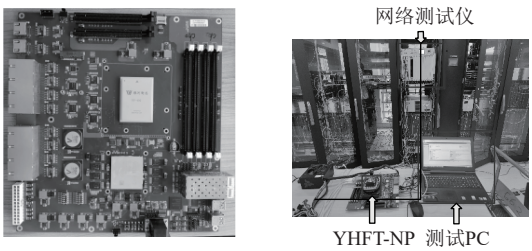
ChipletNP 可重构分组处理流水线为用户提供清晰的软硬件开发接口来屏蔽底层的设计细节，例如，可以屏蔽报文收发、软硬件报文交互等设计细节，从而降低软硬件功能的开发难度. 用户开发网络功能时，需要遵循 3 类相关的开发接口规范，详见附录 A. 用户不但可以根据需求动态增加或删除软件功能模块，而且可以通过离线加载的方式重组硬件功能模块，满足了用户对网络功能灵活可扩展的需求.

5 实验评估与功能验证

5.1 性能测试

为了测试 YHHX-NP 性能，搭建实验环境，对 NP 原型芯片相关性能进行测试. 图 7 所示为 YHHX-NP 芯片性能测试平台. 芯片验证板通过串口连接测试 PC，10 Gbps/40 Gbps 以太网接口通过光纤连接网络测试仪，并在通用多核 CPU 芯粒上运行操作系统. 基于该性能测试平台，对 NP 原型芯片的网络分组转发性能、处理延迟等能力进行验证.

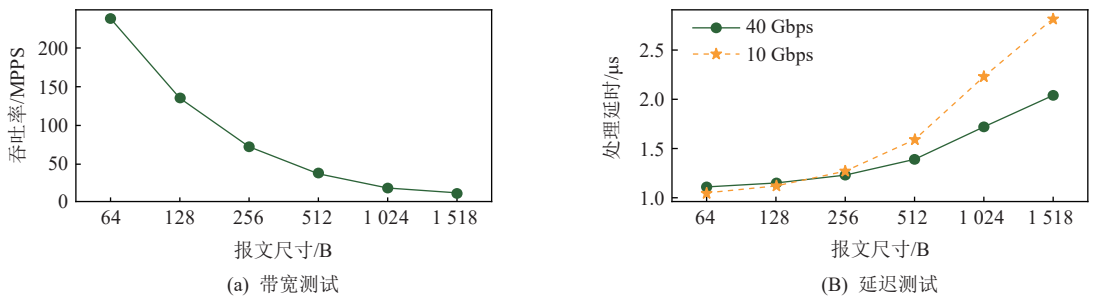
针对不同大小报文的吞吐率测试结果表明，如图 8(a)所示，NP 芯片可满足 10 Gbps/40 Gbps 以太网接口 IPv4, IPv6 转发报文线速处理需求. 如图 8(b)所



(a) 原型芯片验证板 (b) 原型芯片性能测试环境

Fig. 7 YHHX-NP platform for performance test

图 7 YHHX-NP 性能测试平台



(a) 带宽测试 (B) 延迟测试

Fig. 8 Performance test based on prototype system

图 8 基于原型系统的性能测试

示, 处理延迟实验表明, 随着报文尺寸的增大, 交换转发处理延迟有所增加, 但最大不超过 2.82 μs. 查表中关键字通常不超过 96 B(64 B+32 B), 查表结果不超过 32 B. 根据延迟测试结果, 交换转发处理延迟通常可控制在 1.2 μs 延迟以内(报文长度在 256 B 以内). 大报文分组在 40 Gbps 速率下转发延迟优于 10 Gbps, 原因在于 40 Gbps 链路传输速率更快, 而小报文则反之, 原因在于 40 Gbps 的报文分组处理逻辑更长, 处理小报文分组转发时, 分组数据在更长的处理逻辑路径上执行, 延迟稍大.

5.2 典型业务部署验证

5.2.1 SRv6 功能映射

ChipletNP 架构能够对各类新型网络协议提供快速部署的支持. 以 SRv6 为例, 通过对 SRv6 的处理流程进行抽象并对功能进行模块化解耦, 可将 SRv6 的功能实现完全映射到 YHHX-NP 芯片中. 本节将演示基于 YHHX-NP 的 SRv6 显示路由功能.

SRv6 路由器是通过 SRv6 技术实现可配置路由路径和可编程网络的 IPv6 路由器. 基于 YHHX-NP 实现 SRv6 处理引擎功能设计, 及实现显式路径, SRv6 路由器具有 2 个功能:

1)支持普通 IPv6 报文的路由转发.

2)支持 SRv6 引流(源节点压入 SRH(segment routing head))、SRH 解析和处理(如转发节点将 SID(segment identifier)填入目的 IP)、SRH 弹出(末端转发节点删除 SRH)等 SRv6 相关功能.

为支持 SRv6 网络协议的快速部署, 数据平面需完成分组分类、流分类、SID 分类、路由转发、SRv6 SRH 源端压入、SRv6 SID 解析和处理、目的 MAC 地址替换、最大跳数限制(hop limit)判断和修改、段剩余值 *SL*(segment left)判断和修改、ICMP(Internet control message protocol)代理、统计计数、动作处理等功能. 控制平面需完成生成分组分类表、流分类表、SID 分类表、SRH 压入表、转发表、邻居表, 完成对数据平面的配置功能, 如图 9 所示. 数据平面的报文处理按功能模块解耦后的流程描述见附录 B.

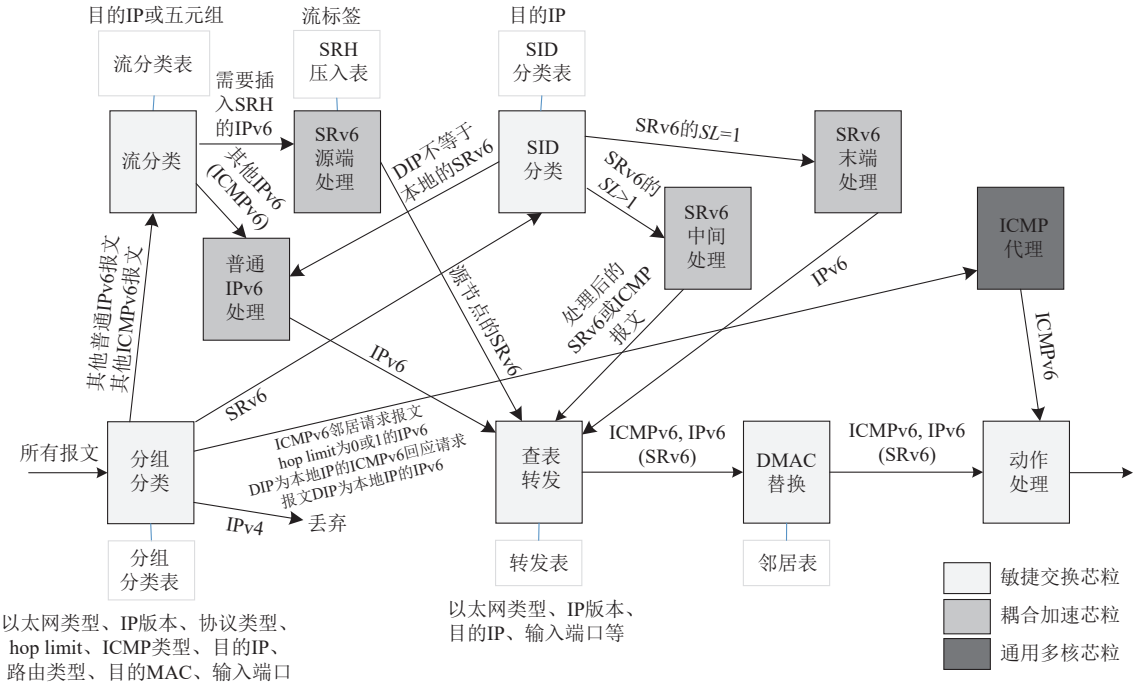


Fig. 9 Function deployment of SRv6 on YHHX-NP prototype chip

图 9 SRv6 在 YHHX-NP 原型芯片上的功能部署

图 9 中敏捷交换芯粒上的映射实现分类查表及简单动作的处理, 包括分组分类、流分类、SID 分类、路由转发、目的 MAC 地址替换、动作处理等功能模块; 耦合加速芯粒上执行需要定制更新报文头字段等较为复杂的动作处理, 包括普通 IPv6 处理、报文解析、SRv6 源端压入 SRH、SRv6 中间处理和末端 SRH 弹出处理、报文汇聚等功能模块. 在通用多核 CPU 阵列芯粒上执行实现 ICMP 代理和规则配置等软件功能模块. 各模块的具体功能描述见附录 C.

5.2.2 SRv6 功能验证

3 台基于 YHHX-NP 芯片实现的路由器 R1, R2, R3 组成一个环形网络, 各路由器之间通过以太网口

3 和以太网口 4 互连. 每台路由器的以太网口 1 分别连接 1 台主机用于收发用户流量. 同时, 为了便于观察路由器之间的报文内容, 将路由器的以太网口 2 作为以太网口 3 和以太网口 4 口的镜像端口, 连接到主机的另外一个外置网口上. 主机 1、主机 2、主机 3 的 IP 地址均不在同一网段. 主机 PC1 的网口连接到 YHHX-NP 路由器 R1 的以太网口, 且这 2 个网口在同一网段, 具体拓扑连接如图 10 所示.

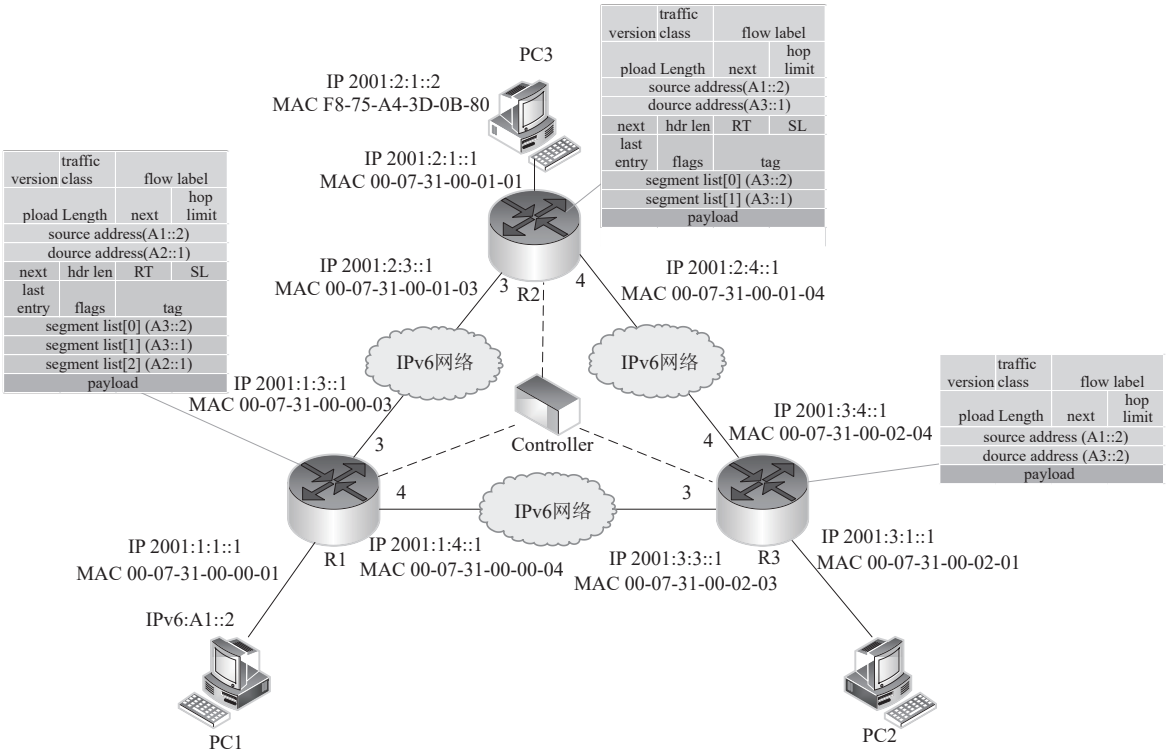


Fig. 10 Topography schematic diagram of SRv6 demo environment

图 10 SRv6 演示环境拓扑示意图

按照网络拓扑实现主机之间通信, 普通 IPv6 报文经过 2 跳路由转发即可. 如主机 1→路由器 1→路由器 3→主机 3. 为了验证基于 YHFT-NP 成功部署 SRv6 显示路由, 本文规划了相应的 SRv6 引流策略和 SRv6 转发路径, 从而实现主机 1 发给主机 3 的报文经过路由器 1、路由器 2、路由器 3 的指定路径, 主机 3 发给主机 1 的报文会经过路由器 3、路由器 2、路由器 1 的指定路径. 本文测试使用抓包软件 Wireshark、串口连接工具 SecureCRT、网络性能测试软件 Iperf3.

如图 11 可以看出路由器 1 收到目的 IP 地址为主机 3(2001:3:1::2)的 ICMPv6 回应请求报文后, 根据 SRv6 引流策略, 成功在报文扩展头中压入段路由头, 段路由携带的段路由列表(SID)为{路由器 2(2001:2:3::1)→路由器 3(2001:3:4::1)→主机 3(2001:3:1::2)}. 此时路由器 1 作为源节点, 根据压入 SRH 中段的剩余值找到相应段路由列表, 即路由器 2 的 IP, 替换 IPv6 目的地址, 并根据新 IPv6 目的地址进行路由转发, 从而将报文发给下一跳路由器 2.

路由器 2 收到报文后, 检查 SRH 头中的 $SL>1$ 作为中间节点, 根据 SRH 中 SL 减 1 后找到相应段路由列表, 即路由器 3 的 IP 地址替换 IPv6 的目的地址, 如图 12 所示, 同时将 $SL-1$ 根据新 IPv6 目的地址进行路由转发, 从而将报文发给路由器 3. 路由器 3 收到报文后, 检查 SRH 头中的 $SL=1$ 作为尾节点, 获取下一跳为主机 3, 并将 SRH 从报文中弹出, 从而将原始报文发送给主机 3.

结果表明, 原型芯片能够成功部署 SRv6 功能, 支持主机 1 发与主机 3 之间的报文按照 SRv6 指定路径进行路由转发, 实现加 SRH 头和去 SRH 头. 通过修改 SRH 的段路由列表实现修改报文转发路径, 能够按需进行新业务的逻辑编程.

5.2.3 资源评估

在本文实验中, 使用国产 FPGA 综合工具评估了映射在耦合芯粒中的关键模块硬件资源利用率, 包括芯粒间通信相关的处理逻辑 FPGA OS、可重构分组处理流水线 2 大部分. 表 2 列出了耦合芯粒中实现

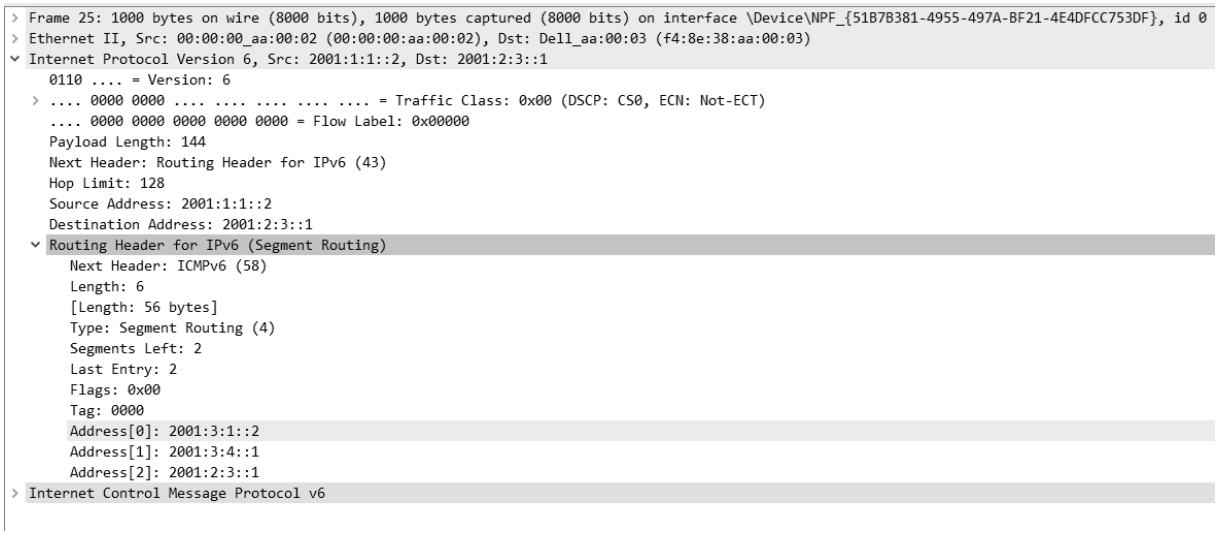


Fig. 11 Packet capture screenshot for SRv6 source end processing

图 11 SRv6 源端处理抓包截图

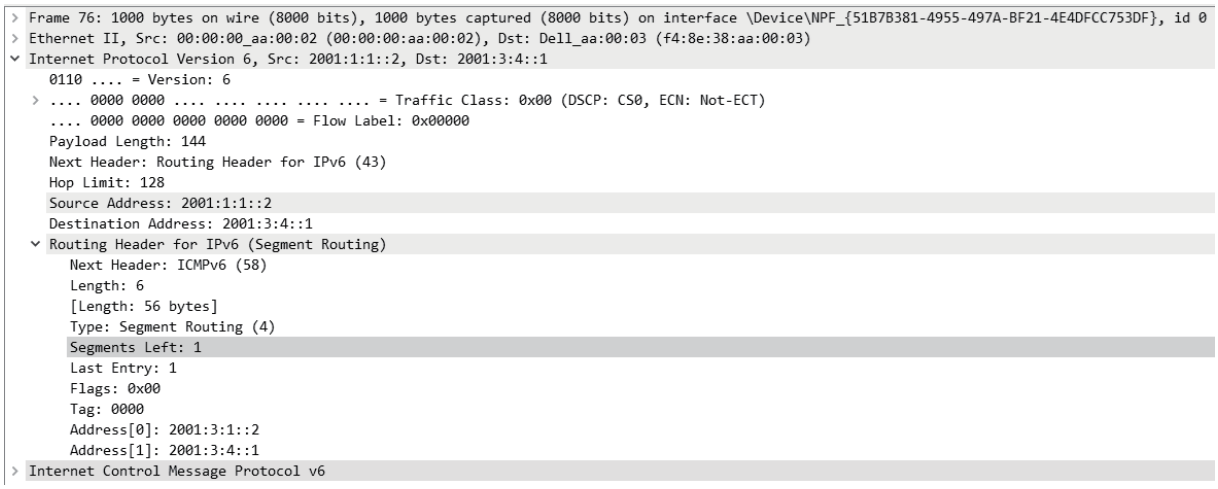


Fig. 12 Packet capture screenshot for SRv6 middle processing

图 12 SRv6 中间过程抓包截图

Table 2 Resource Evaluation of Functional Modules in Coupled Chiplet

表 2 耦合芯粒中功能模块资源评估

SRv6 功能模块	资源		
	LUT	FF	BRAM
普通 IPv6 处理	158	142	0
SRv6 源端处理	2 132	878	5
SRv6 中间处理	1 827	1 213	10
SRv6 末端处理	1 912	1 300	10
解析与汇聚逻辑	3 834	2 932	52
SRv6 功能模块总和	9 863	6 465	77
FPGA OS	15 160	8 461	44
FPGA OS 模块总和	145 016	217 524	526

的功能模块.

工作在 125 MHz 时钟频率下. 硬件资源使用情况, 包括查找表(lookup table, LUT)资源, 即逻辑处理资源; 触发器(flip flop, FF)资源; 块存储(block memory, BRAM)资源. 从表 2 中可以看出, 在耦合芯粒上部署 SRv6 功能模块, SRv6 功能逻辑资源占比约 6.8%, 寄存器资源占比约 2.97%; FPGA OS 功能逻辑资源仅占比约 10.5%, 寄存器资源占比约 3.89%, 可以有效支持新耦合加速功能的部署.

6 结 论

本文提出新型敏捷可定制 NP 架构 ChipletNP, 提出了基于 Chiplet 技术解耦异质资源和交换, 处理、

加速耦合等芯粒的ChipletNP架构以及分组处理和控制模型，定义并设计实现了面向芯粒间统一互连的敏捷交换网络；提出面向网络应用的耦合加速技术，并设计开发了一款集成商用CPU、FPGA和自研敏捷交换芯粒的银河衡芯敏捷NP芯片YHHX-NP，基于该芯片的应用部署与实验结果表明，ChipletNP可支持NP的快速敏捷定制，可以有效承载SRv6等新型网络协议与网络功能部署以满足不同的应用及部署场景，能够支持NP的快速定制和演化发展需求.

作者贡献声明：李韬和杨惠提出了架构设计思路和撰写论文；杨惠提出实验方法并修改全文；厉俊男负责撰写论文部分内容；刘汝霖负责完成实验；孙志刚提出指导意见.李韬和杨惠为共同第一作者.

参 考 文 献

[1] Cisco. A 400Gbps multi-core network processor[EB/OL]. [2023-08-22].https://hc29.hotchips.org/

[2] NXP. QorlQ T2080 and T2081 multicore communications processors [EB/OL]. [2023-02-14].https://www.nxp.com/products/processors-and-microcontrollers/power-architecture/qoriq-communication-processors/t-series/qoriq-t2080-and-t2081-multicore-communications-processors:T2080

[3] Netronome. ODSA Workshop report: Working toward an open multi-Chiplet architecture[EB/OL]. [2023-02-04].https://www.netronome.com/blog/odsa-workshop-report/

[4] Nvidia. Mellanox indigo NPS-400 network processor[EB/OL]. [2023-07-06].https://static6.arrow.com/aropdfconversion/f2ecdca300eb23bee6f2fce7da333eb3659de7f2/pb_indigo_nps-400.pdf

[5] Broadcom. XLP900 multicore, multithreaded, third-generation processor[EB/OL]. [2023-08-04].https://www.digchip.net/datasheets/5897973-xlp900-xlp-ii-processor-family.html

[6] Marvell. Marvell introduces new xelerated network processors and traffic management solutions for the mobile Internet[EB/OL]. [2023-09-01].https://cn.marvell.com/company/newsroom/marvell-introduces-new-xelerated-network-processors-and-traffic-management-solutions-for-the-mobile-internet.html

[7] Marvell. MARVELL OCTEON TX CN82XX and CN83XX[EB/OL]. [2023-04-01].https://www.marvell.com/content/dam/marvell/en/public-collateral/embedded-processors/marvell-infrastructure-processors-octeon-tx-cn82xx-cn83xx-com-product-brief_2019.pdf

[8] NXP. QorIQ Layerscape 2088A and 2048A multicore communications processors[EB/OL]. [2023-04-01].https://www.nxp.com/products/processors-and-microcontrollers/arm-processors/layerscape-communication-process/qoriq-layerscape-2088a-and-2048a-multicore-communications-processors:LS2088A

[9] Qualcomm. Internet processor (IPQ) for enterprise and carrier Wi-Fi products[EB/OL].[2023-04-01].https://www.qualcomm.com/products/ipq8069

[10] Zhao Yuyu, Cheng Guang, Liu Xuhui, et al. Survey and applications of next generation network processor[J]. Journal of Software, 2021, 32(2): 446−474(in Chinese)
(赵玉宇, 程光, 刘旭辉, 等. 下一代NP及应用综述[J]. 软件学报, 2021, 32(2): 446−474)

[11] Sayali D, Hardika S, Niranjan K. High performance packet capturing using data plane development kit[J]. International Journal of Innovative Research in Computer and Communication Engineering, 2018, 6((2):): 6102−6106

[12] Intel. Simulate Intel's next generation communication platform data plane solutions[EB/OL]. [2023-04-01].https://download.intel.com/newsroom/kits/nextgencomm/pdfs/Crystal-Forest_PressDeck.pdf

[13] Barach D, Linguaglossa L, Marion D, et al. High-speed software data plane via vectorized packet processing[J]. IEEE Communications Magazine, 2018, 56(12): 97−103

[14] Li Bojie, Tan Kun, Luo Layong, et al. ClickNP: Highly flexible and high performance network processing with reconfigurable hardware [C/OL]//Proc of the 2016 ACM SIGCOMM Conf. New York: ACM, 2016[2023-09-17].https://www.microsoft.com/en-us/research/wp-content/uploads/2016/07/main-4.pdf

[15] Yin Jieming, Lin Zhifeng, Kayiran O, et al. Modular routing design for Chiplet-based systems[C]//Proc of the 45th ACM/IEEE Annual Int Symp on Computer Architecture. Piscataway, NJ: IEEE, 2018: 726−738

[16] AMD. AMD fully discloses zeppelin soc architecture details at ISSCC 2018-7nm EPYC "Rome" chips rumored to feature up to 64 cores[EB/OL]. [2023-08-15].https://wccftech.com/amd-zeppelin-soc-isscc-detailed-7nm-epyc-64-cores-rumor/

[17] Intel. Tofino 2: Second-generation p4-programmable Ethernet switch ASIC that continues to deliver programmability without compromise [EB/OL]. [2023-08-15].https://www.intel.com/content/www/us/en/products/network-io/programmable-ethernet-switch/tofino-2-series.html

[18] Chole S, Fingerhut A, Ma S, et al. dRMT: Disaggregated programmable switching[C/OL]//Proc of the Conf of the ACM Special Interest Group on Data Communication. New York: ACM, 2017[2023-09-17].https://dl.acm.org/doi/pdf/10.1145/3098822.3098823

[19] Firestone D, Putnam A, Mundkur S, et al. Azure accelerated networking: SmartNICs in the public cloud[C]//Proc of the 15th USENIX Symp on Networked Systems Design and Implementation. New York: ACM, 2018: 51-66

[20] NVIDIA. CONNECTX-5: Advanced offload capabilities for the most demanding applications[EB/OL]. [2023-08-15].https://www.nvidia.com/en-us/networking/ethernet/connectx-5/

[21] Marvell. Marvell LiquidIO II 10/25G SmartNIC family[EB/OL]. [2023-08-15].https://www.marvell.com/content/dam/marvell/en/public-collateral/ethernet-adaptersandcontrollers/marvell-ethernet-liquidio-ii-cn23xx-10g-25g-product-brief-2017.pdf

[22] Huawei. Huawei IN300 FC HBA card user guid[EB/OL]. [2023-08-

15].https://support.huawei.com/enterprise/en/doc/EDOC1100063074/218b0b44

[23] NVIDIA. What is a DPU[EB/OL]. [2023-05-20].https://blogs.nvidia.com/blog/2020/05/20/whats-a-dpu-data-processing-unit

[24] Intel. Intel Infrastructure processing unit (Intel IPU) and SmartNICs [EB/OL]. [2023-08-15].https://www.intel.cn/content/www/cn/zh/products/network-io/smartnic.html

[25] Intel. Intel Infrastructure processing unit (Intel IPU) soc (codename: mount evans)[EB/OL]. [2023-08-15].https://www.intel.sg/content/www/xa/en/products/platforms/details/mount-evans.html?countrylabel=Asia%20Pacific

[26] Li Junnan, Yang Xiangrui, Sun Zhigang. DrawerPipe: A reconfigurable packet processing pipeline for FPGA[J]. Journal of Computer Research and Development, 2018, 55(4): 717−728 (in Chinese)
(厉俊男, 杨翔瑞, 孙志刚. DrawerPipe: 基于 FPGA 的可重构分组处理流水线模型[J]. 计算机研究与发展, 2018, 55(4): 717−728)

[27] Anwer M B, Motiwala M, Tariq M, et al. SwitchBlade: A platform for rapid deployment of network protocols on programmable hardware [C]//Proc of the ACM SIGCOMM Conf. New York: ACM, 2010: 183−194

Li Tao, born in 1983. PhD, associate professor. His main research interest includes network architecture design.

李 韬, 1983 年生. 博士, 副研究员. 主要研究方向为网络体系结构设计.

Yang Hui, born in 1986. PhD, associate professor. Member of CCF. Her main research interest includes network processor .

杨 惠, 1986 年生. 博士, 副研究员. CCF 会员. 主要研究方向为网络处理器.

Li Junnan, born in 1992. PhD, assistant professor. His main research interests include network router architecture and IC design.

厉俊男, 1992 年生. 博士, 助理研究员. 主要研究方向为网络路由器体系结构、集成电路设计.

Liu Rulin, born in 1989. PhD, assistant professor. His main research interests include network architecture and IC design.

刘汝霖, 1989 年生. 博士, 助理研究员. 主要研究方向为网络体系结构、集成电路设计.

Sun Zhigang, born in 1974. PhD, professor. His main research interests include software-defined network, time-sensitive networking, network architecture, and network security.

孙志刚, 1974 年生. 博士, 研究员. 主要研究方向为软件定义网络、时间敏感网络、网络体系结构、网络安全.

附录 A. 可重构分组处理流水线开发接口规范

如图 A1 所示为可重构分组处理流水线开发接口规范：

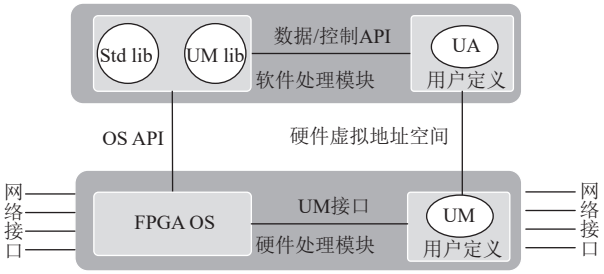


Fig. A1 Interface specification for reconfigurable packet processing pipeline development

图 A1 可重构分组处理流水线开发接口规范

1)UM 接口(user module interface)规范. 将耦合芯粒的功能划分为 2 部分, 即芯粒间通信相关的处理逻辑 FPGA OS 和可重构分组流水线中的用户自定义硬件功能模块 UM. 因此, 使用 UM 接口规范定义 UM 逻辑与 FPGA OS 间的接口信号和工作时序, 其中接口信号包括 UM 从 FPGA OS 接收报文、向 FPGA OS 发送报文、接收软件模块发送的配置消息, 以及调用 FPGA OS 提供的其他功能.

2)数据/控制 API 定义了软件 UA(user applications)访问底层资源使用的关键数据结构和相关函数. 包含 2 个部分: ①UA 开发提供了分组收发, 软件、硬件规则管理配置等通用 API, 标准 API 的库函数(Std lib). ②可重构分组流水线的用户自定义硬件功能模块 UM 增加一些特殊的处理功能, 这些自定义硬件模块的管理软件 API, 即 UM 相关库(UM lib).

3)硬件虚拟地址空间(virtual address space). 硬件地址空间规范定义了软件可访问 FPGA OS 和 UM 的硬件资源, 如寄存器、计数器和存储器等, 并为所有可以访问的存储单元定义了唯一的虚拟地址供软件 UA 访问. 唯一的硬件地址空间是 UA 和 UM 跨硬件平台可移植的前提.

附录 B. SRv6 数据平面报文处理

1)外部网络报文进入分组分类模块后, 分组分类模块通过查询分组分类表可实现对报文初步的分类. 若是 IPv4 报文, 则直接丢弃; 若是 hop limit 为 0 或 1 的报文、ICMPv6(ICMP over IPv6)邻居请求或 ping 本机报文, 则送给 ICMP 代理模块; 若是 SRv6 报文, 则送给 sid 分类模块; 其他普通的 IPv6 报文, 则送给流分类模块.

2)普通的 IPv6 报文进入流分类模块后, 流分类模块通过查流分类表判断报文是普通 IPv6 处理还是 SRv6 加头处理. 若报文目的 IP 命中, 说明该流需要进行 SRv6 加头处理, 则送给 SRv6 源端处理模块; 若报文目的 IP 未命中, 说明该流需要进行普通 IPv6 处理, 则送给普通 IPv6 处理模块.

3)SRv6 报文进入 SID 分类模块后, SID 分类模块查 SID 分类表判断是否需要本机做 SRv6 处理, 还是作为普通 IPv6 报文处理. 若需要本机做 SRv6 处理, 再进一步判断是 SRv6 中间处理还是 SRv6 末端处理模块. 若报文目的 IP 命中, 说明该流需要本机做 SRv6 处理. 再判断 SRH 中的 *SL* 字段, 若 *SL*=1 则送给 SRv6 末端处理模块; 若 *SL*≠1 则送给 SRv6 中间处理模块; 若报文目的 IP 未命中, 说明该流需要进行普通 IPv6 处理, 则送给普通 IPv6 处理模块.

4)转发分组报文进入查表转发模块后, 查表转发模块通过查转发表获取转发分组的输出端口号、邻接表索引 SRv6 源端处理模块、SRv6 中间处理模块、SRv6 末端处理模块、普通 IPv6 处理模块, 这些模块处理转发分组, 并且转发分组处理完之后需要进行路由转发.

5)转发分组进入目的 MAC 替换模块后, 目的 MAC 替换模块通过查邻接表获取目的 MAC 且完成原报文的替换.

6)转发分组进入动作处理模块后, 动作处理模块完成转发分组源 MAC 的替换.

7)ICMP 代理模块处理非转发模块, 这类报文只需要沿原路返回即可, 并不需要进行路由转发. 因此 ICMP 代理构造好相应的报文之后直接由动作处理模块按照指定输出端口转发即可.

附录 C. SRv6 数据平面功能模块

1)分组分类模块. 通过查找分组分类表实现对报文分组的初步分类, 把 IPV4 报文丢弃; 把 hop limit 为 0 或 1、ICMPv6 邻居请求报文、ICMPv6 ping 本机报文送 CPU 的 ICMP 代理模块; 把 SRv6 报文送入 SID 分类模块; 把普通 IPv6 报文送入流分类模块.

2)流分类模块. 通过查流分类表判断报文是否需要增加 SRv6 的头. 使用目的 IP 字段作为判断依据, 仅目的 IP 命中的报文才需要增加 SRv6 的头, 未命中

的报文则做普通 IPv6 处理.

3)SID 分类模块. 处理 SRv6 报文时通过查 SID 分类表判断报文是否需要本机做 SRv6 相关处理，以及判断是 SRv6 中间处理还是末端处理. 使用目的 IP 字段和 SRH 中的 *SL* 字段作为判断依据，仅目的 IP 命中的报文才需要本机做 SRv6 相关处理. 而 SRv6 的处理需要依据 SRH 中的 *SL* 字段进一步判断是 SRv6 中间处理模块还是 SRv6 末端处理模块.

4)路由转发模块. 通过查找转发表实现路由信息的获取. 根据目的 IP、目的 IP 掩码进行查表，获取输出端口号和路由表项索引，查表结果可携带给下一级流表.

5)目的 MAC 替换模块. 目的 MAC 替换模块根据路由表项索引进行查表，获取目的 MAC 地址并对报文进行替换.

6)动作处理模块. 可对报文的源 MAC 进行替换，也可根据指定输出端口对报文进行转发.

7)普通 IPv6 处理模块. 对于普通正常的 IPv6 报文，修改报文 VLAN 标示中的目的标示.

8)SRv6 源端处理模块. 实现 SRv6 路由器源端节点功能，包括根据流 ID 查表压入 SRH，更新 IPv6 报文头中目的 IP、报文长度，以及 SRH 中下一跳地址.

9)SRv6 中间处理模块. 实现 SRv6 路由器中间节点功能，包括对报文目的 IP 进行修改，以及将 SL 减 1.

10)SRv6 末端处理模块. 实现 SRv6 路由器末端节点功能，包括弹出 SRH、将 IPv6 头中的下一跳地址替换为 SRH 首部中的地址、将 IPv6 头中报文长度减去 SRH 总长度.

11)ICMP 代理模块. 为邻居请求、hop limit 为 0 或 1、MTU 请求等报文构造相应的 ICMPv6 响应报文.

12)流表配置模块. 将控制信息转换成流表规则并配置到交换芯片. 仅当流表配置模块完成对交换芯片的配置之后，才能对数据报文进行处理.

13)SRv6 报文汇聚模块. 采用轮询和效率调度方式将 SRv6 源节点报文、SRv6 中间节点报文、SRv6 尾节点报文和 IPv6 报文数据合并成一路输出.

14)SRv6 报文解析模块. 对报文数据进行分类解析，将其区分成普通 IPv6 报文、源节点 IPv6 报文、中间节点 SRv6 报文、尾节点 SRv6 报文，并生成相应解析结果. 各类报文和解析结果分别输出到不同下游模块.