智能网卡：云计算时代的高新网络技术

陈 果[1] 李肯立[1] 罗腊咏[2] 谭 焜[3]
[1] 湖南大学
[2] 腾讯 TEG
[3] 华为中央软件院

关键词：数据中心网络 智能网卡 硬件网络协议栈

数据中心是云计算时代最重要的基础设施之一，为云提供强大的计算和存储能力。数据中心网络作为联合数据中心内所有计算、存储单元的“交通枢纽”，是影响云服务能力的关键所在。然而，近年来业界对数据中心网络性能以及虚拟化能力的需求日益增长，给现有端系统中网络协议栈的处理带来了极大挑战。一方面，目前业界普遍的百 G 级带宽、微秒级延时的高性能网络设施需要端系统协议栈具备极高的处理速度，因此将协议栈的处理卸载 (offload) 到网卡 ASIC 芯片上实现是目前普遍采用的方式；另一方面，为应对层出不穷的虚拟化需求，网络协议栈所包含的处理功能也需要频繁更新，这又使将协议栈的处理卸载到网卡 ASIC 芯片上变得非常困难。在此背景下，智能网卡 (smart NIC) 技术开始走进人们的视野并逐渐得到普及。不同于传统网卡，智能网卡同时具备高性能及可编程的能力，既能处理高速的网络数据流，又能对网卡进行编程，实现定制化的处理逻辑。

网卡的发展史

网卡是端系统接入网络进行通信的必备设备，端系统 CPU 和网卡联合完成整个网络协议栈中各层的处理。

传统网卡

早期的网卡仅实现数据链路层和物理层的功能，而端系统 CPU 负责处理网络协议栈中更高层的逻辑。CPU 按照网络协议栈中传输层、路由层的逻辑，负责数据包的封装和解封；网卡则负责更底层的数据链路层帧的封装和解封，以及物理层电气信号的相应处理。

高性能网卡

链路带宽的增长对端系统协议栈的处理速度提出了更高的要求。例如，要想在 100G 以太网中线速处理 64 字节大小的数据包，只有几个时钟周期供 CPU 进行网络协议栈的处理。为适应高速网络，现代网卡硬件中普遍卸载了部分传输层和路由层的处理逻辑（如校验和计算、传输层分片重组等），来减轻 CPU 的处理负担。甚至有些网卡（如 RDMA 网卡）还将整个传输层的处理都卸载到网卡硬件上，以完全解放 CPU。得益于这些硬件卸载技术，端系统的网络协议栈处理才能与现有的高速网络相匹配。

智能网卡的兴起

近年来公有云中网络虚拟化的发展以及 host-based SDN 技术的兴起 [1]，对端系统协议栈提出了更高的要求，而传统的高性能网卡已难以满足这些要求。

当前的云服务提供商除了出售虚拟机服务外，还提供“基础设施即服务 (Infrastructure-as-a-Service, IaaS)”这种服务模式。IaaS 需要支持各式各样的网络功能，例如可让用户任意配置 IP 的虚拟私有网络、可扩展的 4 层负载均衡器、安全策略和访问控制列表、带宽计量以及服务质量控制等。这些网络功能目前普遍采用软件定义网络 (Software Defined Network, SDN) 的方式在端系统的协议栈内实现，端系统需在 TCP/IP 协议栈之外对数据包实现进一步的封装 / 解封装、计数、丢弃等操作。因为用户对于新网络功能的需求层出不穷，SDN 协议栈支持的网络功能往往以月为单位频繁更新迭代 [1,2]。ASIC 芯片的生产从确定需求到最后流片量产往往需要数年的时间，网卡厂商几乎不可能预测到未来几年内所有可能出现的网络功能，因此难以将这些处理逻辑实现到网卡的 ASIC 芯片上。

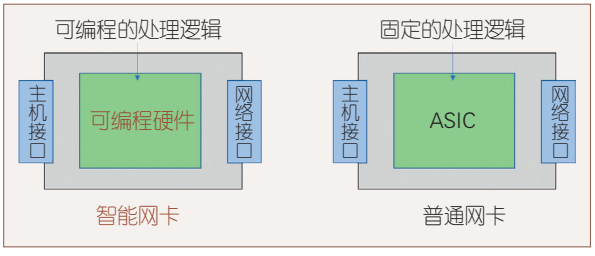

图 1　智能网卡与普通网卡的结构对比

在此背景下，智能网卡这项新兴技术开始走进人们的视野 [2~5]。智能网卡是一种可编程的高性能网卡。如图 1 所示，智能网卡上具备可编程硬件，可对网卡的处理逻辑进行动态的编程重写。不同于普通网卡中通过 ASIC 实现固定的网络处理逻辑，智能网卡兼具高速的处理能力及灵活的可编程能力。

现在典型的智能网卡有两种实现方式，最典型的一种是采用**现场可编程门阵列** (Field-Programmable Gate Array, FPGA) 实现。微软公司是这方面的代表 [2]，已在其 Azure 云内面向用户提供服务，腾讯公司也在积极研制自己的基于 FPGA 的智能网卡 [4]。另一种智能网卡的实现方式是采用**专用的多核网络处理器** (Network Processor, NP)。华为公司采用的正是这类智能网卡 [3]。

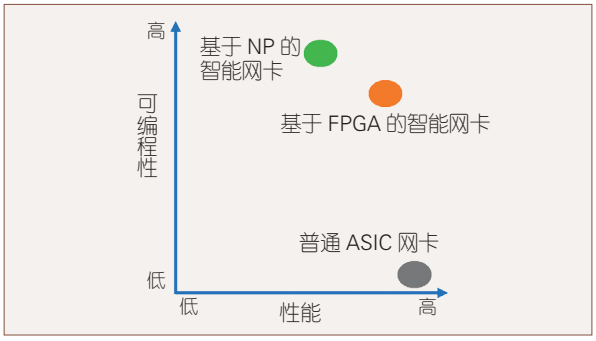

图 2　典型智能网卡实现方式的可编程性以及处理性能对比

如图 2 所示，这两种智能网卡的实现方式在可编程性以及处理性能上各有优劣。在性能方面，通过直接烧写硬件逻辑，基于 FPGA 的智能网卡性能与 ASIC 芯片非常接近；NP 则采用多核的方式来提高整个网卡的处理能力，但其每个核的处理能力与通用 CPU 性能相比并无特别优势，因此对于单一网络流的处理性能较低。在可编程性方面，基于 FPGA 的智能网卡可编程性相对较低。一方面因为 Verilog/VHDL 语言编程难度大，并且开发 FPGA 的整套工具链的易用性也有待提高；另一方面因为 FPGA 受限于芯片面积，往往无法实现较复杂的处理逻辑。而 NP 则具备较高的可编程性。首先，其编程多采用通用的 C/C++ 语言，相对简单；其次，NP 具备几乎和通用 CPU 相当的表达能力，可实现灵活的处理逻辑。

智能网卡的应用场景与研究热点

网络协议栈卸载

智能网卡的兴起来自对网络虚拟化的需求及对相关协议栈高速处理的需求，故协议栈的卸载是目前智能网卡最典型的应用。以微软公司和腾讯公司

为代表，这些云厂商主要将其网络中的 SDN 协议栈实现到智能网卡中 [2,4]，从而减轻甚至消除网络虚拟化处理中 CPU 的负担；而华为公司除了利用智能网卡处理网络虚拟化外，还将现有网络协议栈中的全部传输层逻辑（如 TCP）卸载到网卡中进行高速处理 [3]，从而进一步加速网络传输。

将协议栈卸载到网卡硬件中实现与传统的软件实现环境有很大不同。网卡的片上存储空间要远远小于软件环境中可以使用的主机内存大小，这就需要设计新的体系架构和算法来实现高效的硬件协议栈。具体来说：

- 减小 SDN 协议栈的存储开销一般可以通过动态流表的方式实现 [2]。网卡芯片上仅缓存目前活跃的少量数据流的处理表项，而其他数据流的处理表项都存储在片外 DRAM 或主机内存中。当新流建立时，CPU 会查询片外的 SDN 流表，明确相应处理步骤，再将流表下发至网卡芯片，由其处理该流后续的数据包。
- 在网卡硬件中实现传输层逻辑则更加复杂。目前的传输层协议中包括拥塞控制算法、丢包恢复处理等在内的各方面逻辑都需要消耗较多存储，不利于硬件实现。近年来诞生了一些新的传输层算法来解决此问题，笔者也在这方面做了一些工作。例如，MPRDMA[6] 设计了一套低存储开销的多路径传输层协议，并基于微软的智能网卡进行了实现。该方案可使用与路径数量无关的常数存储开销实现多路径拥塞控制。MELO[7] 则设计了一套低存储开销的选择重传丢包恢复机制，可显著降低选择重传中需维护的相关数据结构的存储开销。

网络应用加速

在处理数据包之外，具备可编程能力的网卡还能对上层应用的逻辑进行一些加速处理。因此，智能网卡的第二类典型应用就是对上层网络应用的加速，近年来有不少工作在这方面做出了成果。例如，KV-Direct[8] 通过在智能网卡上实现键值存储数据库 (Key-Value Store, KVS) 的键值操作，可跳过 CPU，利用网卡直接读取或更新远端主机内存中的键值，从而将单机的键值操作吞吐量提升到 10 亿次 / 秒的级别。文献 [9] 则使用智能网卡对数据压缩算法进行加速，并在微软的 Azure 云存储服务中得到了应用。

离散化数据中心

传统的数据中心是以服务器为单位构建的高速互连的服务器集群，其中每台服务器都具备少量的计算和存储资源（如 CPU、内存、硬盘等）。近期业界开始探索一种新型的架构范式——离散化数据中心 (disaggregated data center)，试图以资源为单位将数据中心构建为一个大型计算和存储的资源池。离散化数据中心打破服务器的限制，利用高速网络将相同甚至不同机柜内的多台服务器中的同类型资源直接互联为一个整体，构建 CPU 资源池、内存资源池、存储资源池、GPU 资源池，等等。支持资源池的灵活访问需要更高的网络带宽以及更低的网络延迟，这给现有的数据中心网络技术带来巨大挑战，而应用智能网卡则很有希望突破这些难点。

利用智能网卡的定制化逻辑，我们无须采用传统的网络通信方式，可以跳过 CPU 的参与，在远程资源之间通过网卡直接进行高速互连。例如，S-Direct[10] 利用智能网卡，将数据中心内的 NVMe 闪存设备进行高速互连，其尾部访问延时低于现有通过主机 CPU 参与互连方式的 1/8。而笔者近期的工作 DUA[11]，则从数据中心内部署的 FPGA 芯片的视角出发，利用智能网卡为 FPGA 提供直接高速访问数据中心内其他所有计算和存储资源的统一通信架构，向离散化数据中心又迈出了重要一步。

挑战与机遇

虽然智能网卡目前已经在各方面取得了成功应用，但仍有一些问题需要业界共同探索解决。

智能网卡架构的选择

目前业界已取得的共识是：在高性能之外，网卡还需要“智能”，即可编程能力。然而现有集成电

路的结构决定了在网卡芯片面积不变的前提下，提升可编程能力必然要在一定程度上牺牲其处理性能。究竟在两者之间如何取舍才能最好地实现满足各种场景的智能网卡，当前尚无定论。当前两种典型的实现方式——基于 FPGA 和基于 NP，在可编程性和处理性能两方面各有优劣，难以兼顾。从理论上来说，若想兼顾高性能和智能，一个最优的智能网卡架构应将不同场景下各网络协议栈都需具备的相同处理逻辑固化成专用 ASIC 芯片，而将其余随场景变化的处理逻辑通过可编程芯片定制化实现。然而现实面对的困难是，目前对网络协议栈各层的功能没有一个很好的模块化抽象，各层内的各个处理逻辑之间甚至跨层的部分逻辑之间都依据场景高度耦合，难以提取有意义的共性和特性逻辑分别由 ASIC 和可编程芯片处理。因此，现有智能网卡要想具备足够的可编程能力，能应对各种场景，只能将从上到下各层几乎所有的逻辑都交由可编程芯片实现，牺牲了性能。例如目前的传输层协议中拥塞控制、丢包恢复、保序传输等逻辑都高度耦合在一起，若想利用智能网卡支持定制化的拥塞控制算法，只能在可编程芯片中实现整个传输层逻辑。

若想突破此难题，一种可行的思路是从模块化的角度出发重构现有网络协议栈，在其各层协议中清晰地抽象出共性功能模块以及可依据场景定制的特性模块。例如，SDN 协议栈中各种功能都可以抽象为流表和匹配操作的模式 [1]，如果能将各 SDN 场景中所采用的匹配操作进行综合总结，则有可能将其固化为几种类型的通用 ASIC 处理逻辑模块，而用户只需根据不同场景对这些逻辑模块进行组合编程，即可实现足够智能的网卡功能。又如，可将传输层协议中的拥塞控制、丢包恢复、保序传输等功能抽象为解耦的独立模块，交由可编程芯片处理，而固化其他共有的传输层处理逻辑。从网卡设计的角度出发，各种场景下的网络协议栈中到底哪些部分可以由 ASIC 固化、哪些部分需要灵活地定制逻辑，值得进一步研究和探索。

分布式系统的性能优化

高性能网卡具备可编程能力，也给分布式系统的性能优化带来了更大机遇。现有的分布式系统设计皆是基于传统的网卡进行，虽然已有不少工作利用智能网卡加速上层应用，但大多是在已有系统架构上寻求可以使用网卡加速的部分。如何在设计整个分布式系统架构之初就把网卡的可编程能力考虑在内，寻求最优的系统性能，值得深入探索。这方面有两层含义：

系统中计算、存储逻辑的合理划分。哪些计算和存储逻辑在网卡上实现，哪些计算和存储逻辑在 CPU 上实现依然能保证最佳的系统性能，是智能网卡时代每一个分布式系统的设计者在设计系统架构时应该仔细考虑的问题。例如在设计分布式机器学习系统时，除了参数服务器和 All-Reduce 模式外，是否可以基于智能网卡的能力进行其他方式的数据存储或计算模型的划分，设计出性能更佳的并行训练模式？又如，设计分布式大数据系统时，除了将底层数据库的存取操作卸载到网卡实现，是否可以考虑将更上层的数据分析语义也一并由网卡实现？这些都值得深入思考。当然，我们更期待出现一个智能网卡背景下分布式系统的设计指导理论，帮助设计者在各种应用逻辑和软硬件环境中选择最优的系统架构。

异构资源下的编程框架。在选定系统架构之后，另一个需要考虑的问题是开发实现的难度。智能网卡背景下的分布式系统编程不再是单纯的 CPU 软件编程，而需要一个更加高效、易用的编程框架让我们更好地对系统中的 CPU、智能网卡硬件，甚至其他异构资源（如 GPU）一起进行协同编程，从而搭建一个高性能的异构分布式系统。该编程框架需考虑编程语言、各种可复用的功能库、运行环境支撑、调试环境等方面，来简化系统的开发。现有的一些工作如 OpenCL[14] 和 ClickNP[12] 在这个方向做出了重要努力，但不论是易用度、性能，还是相关硬件厂商的支持，离大规模应用都还有不小的距离。

更加开放的生态环境

目前智能网卡开发和研究中存在的一个重要问

题是相关的软硬件生态环境还不够成熟。不论是网卡硬件，还是相关的软件工具链，都缺乏能够向普通研发人员开放的友好的研发环境。当前智能网卡的研发力量主要集中在几个具备网卡软硬件全栈研发实力的大公司内部。智能网卡的发展与应用离不开广大研发人员的参与，只有具备足够友好开放的生态环境，才能更加快速地推动一项新技术的发展。令人欣慰的是，如 P4[13] 和 NetFPGA[15] 等工作已在可编程网络硬件的通用生态方面做出了重要努力，期待不久的将来智能网卡这项技术会更加普及与开放。 ■

陈　果

CCF 专业会员。湖南大学信息科学与工程学院副教授。主要研究方向为高性能硬件网络协议栈、数据中心网络、网络系统。guochen@hnu.edu.cn

李肯立

CCF 杰出会员、高性能计算专委常委。湖南大学教授、博士生导师，国家超算长沙中心主任。主要研究方向为高性能计算、并行与分布式处理。
lkl@hnu.edu.cn

罗腊咏

腾讯 TEG 网络平台部专家工程师、云网络系统组组长。主要研究方向为云网络架构、智能网卡和 NFV。
laluo@tencent.com

谭　焜

华为中央软件院副总裁、云网络实验室主任。主要负责终端和云网络软件的开发和创新。
kun.tan@huawei.com

参考文献

[1] Firestone D. VFP: A virtual switch platform for host SDN in the public cloud[C]//*USENIX Symposium on Networked Systems Design and Implementation* (NSDI). 2017: 315-328.

[2] Firestone D, Putnam A, Mundkur S, et al. Azure accelerated networking: SmartNICs in the public cloud[C]//*15th USENIX Symposium on Networked Systems Design and Implementation* (NSDI 18), Renton, WA. 2018.

[3] Huawei released intelligent acceleration engine series to fuel enterprise applications with new speeds[OL]. https://www.huawei.com/en/press-events/news/2018/10/intelligent-acceleration-engine-series.

[4] Luo L. Towards converged SmartNIC architecture for bare metal & public clouds. APNet 2018 Industry Talks. 2018

[5] Mellanox Innova™-2 Flex Open Programmable SmartNIC[OL]. http://www.mellanox.com/page/products_dyn?product_family=276&mtag=programmable_adapter_cards_innova2flex, 2018.

[6] Lu Y, Chen G, Li B, et al. Multi-Path Transport for RDMA in Datacenters[C]//*15th USENIX Symposium on Networked Systems Design and Implementation* (NSDI 18). USENIX Association, 2018.

[7] Lu Y, Chen G, Ruan Z, et al. Memory efficient loss recovery for hardware-based transport in datacenter[C]//*Proceedings of the First Asia-Pacific Workshop on Networking* (APNet). ACM, 2017: 22-28.

[8] Li B, Ruan Z, Xiao W, et al. KV-direct: High-performance in-memory key-value store with programmable NIC[C]//*Proceedings of the 26th Symposium on Operating Systems Principles*. ACM, 2017: 137-152.

[9] Fowers J, Kim J Y, Burger D, et al. A scalable high-bandwidth architecture for lossless compression on fpgas[C]//*2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines*(FCCM). IEEE, 2015: 52-59.

[10]Lu Y. Towards High Performance Disaggregated Flash Storage with Programmable NIC[C]//*The 26th ACM Symposium on Operating Systems Principles, Student Research Competition*(SOSP SRC), October 28-31, 2017.

[11]Shu R, Cheng P, Chen G, et al. Direct universal access: Making data center resources available to FPGA[C]//*16th USENIX Symposium on Networked Systems Design and Implementation* (NSDI 19). 2019.

[12]Li B, Tan K, Luo L L, et al. Clicknp: Highly flexible and high performance network processing with reconfigurable hardware[C]//*Proceedings of the 2016 ACM SIGCOMM Conference*. ACM, 2016: 1-14.

[13]Bosshart P, Daly D, Gibb G, et al. P4: Programming protocol-independent packet processors[J]. ACM SIGCOMM Computer Communication Review. 2014, 44(3): 87-95.

[14]OpenCL Overview[OL]. https://www.khronos.org/opencl/.2018.

[15]NetFPGA[OL]. https://netfpga.org/site/. 2018.