CAInNet:面向 AI 加速的通算一体网内计算模型

刘忠沛 杨翔瑞 杨 凌 高源航 吕高锋 王宝生 苏金树

(国防科技大学计算机学院 长沙 410073)

摘 要 AI 时代的到来对当今算力提出了双重挑战,一方面涉及推理,另一方面涉及分布式训练。将一部分分布式应用的计算任务卸载到高速网络的网卡或交换机能够潜在提升分布式应用的性能表现,并发挥网络的关键作用。如在交换机或网卡中卸载参数聚合等计算功能能够有效降低模型训练时产生的大量通信开销。基于 P4 语言的可编程数据平面除了使网络协议定制更加灵活外,还使得网络数据平面能够为分布式应用提供简单的网内计算服务。然而,当前典型的基于 P4 语言的可编程数据平面架构如协议无关交换架构(PISA)在进行矩阵运算等方面还表现得不够高效。分析该缺陷的关键原因在于:PISA 架构中的超长指令字计算引擎在处理大规模并行同构计算任务时效率不高。针对上述问题,提出了一种面向 AI 加速的通算一体网内计算模型 CAInNet。该模型在传统可编程数据平面的基础上,创新性地融合了单指令多数据流(SIMD)与多指令多数据流(MIMD)两种计算模式,使得网络设备不仅能够支持协议无关网络分组处理,还能在分组传输过程中对承载 AI 推理与训练的数据做网内计算。为了验证 CAInNet 在网内计算以及网络可编程方面的能力和效果,我们在该模型中使用带内网络遥测实现网络可视化,并部署多层感知机(MLP)模型实现基于 AI 的报文分类,替代传统的基于 TCAM 查表的路由方法。实验表明,采用机器学习推理的报文分类方法在包含 5k 路由表项的场景下,其准确度高达 98.3%,同时节省了 98.7%的存储空间,有效地解决了路由爆炸问题。与现有方法相比,将机器学习推理部署在 CAInNet 中不增加可编程数据平面的处理延迟,且仅消耗适量计算资源。

关键词 AI 硬件加速;通算一体;网内计算;可编程网络;报文分类;深度神经网络

中图法分类号 TP393 **DOI 号** 10.11897/SP.J.1016.2025.00019

CAInNet: In-Network Computing Model for AI Acceleration

LIU Zhong-Pei YANG Xiang-Rui YANG Ling GAO Yuan-Hang

LÜ Gao-Feng WANG Bao-Sheng SU Jin-Shu

(College of Computer Science and Technology, National University of Defense Technology, Changsha 410073)

Abstract The operation and service provision of distributed machine learning models are inseparable from computing power and network support. As Moore's Law slows down and the rate of computing power growth is much slower than the rate of I/O, near-data processing has become the inevitable choice in the post-Moore era. In short, it means moving data around as little as possible so that it can be processed on the path. As the transmission path of data, a high-speed network connects multiple computing devices together to form a system that communicates and cooperates with each other. In AI applications, high-speed networks are the bridge connecting algorithms, data, and computing power. The arrival of the era of AI poses two challenges to today's computing power. On the one hand, it involves inference and on the other hand, it involves distributed training. Offloading part of the computing tasks of distributed applications to the

收稿日期:2024-01-16;在线发布日期:2024-09-25。本课题得到国家自然科学基金(62372462)、湖南省自然科学基金(2023JJ40682)、国防科大青年自主创新科学基金(ZK2023-13)与国防科技大学高层次人才资助项目资助。**刘忠沛**,博士研究生,主要研究方向为计算机网络。E-mail:747541120@qq.com。**杨翔瑞**(共同第一作者),博士,助理研究员,主要研究领域为可编程网络。E-mail:yangxiangrui11@nudt.edu.cn。**杨 凌**,博士研究生,主要研究方向为计算机体系结构。**高源航**,硕士研究生,主要研究方向为计算机网络。**吕高锋**,博士,副研究员,主要研究领域为 FPGA 网络加速器。**王宝生**(通信作者),博士,研究员,主要研究领域为网络安全。E-mail:bswang@nudt.edu.cn。**苏金树**,博士,研究员,中国计算机学会(CCF)高级会员,主要研究领域为计算机网络、网络安全。

network cards or switches of high-speed networks can potentially improve the performance of distributed applications and play a key role in the network. Offloading computation functions such as parameter aggregation in switches or network cards can effectively reduce the large amount of communication overhead incurred during model training. The current programmable data plane based on P4 language not only makes the customization of network protocol more flexible, but also makes the network data plane provide simple in-network computing services for distributed applications. However, the typical P4 based programmable data plane architecture such as Protocol Independent Switch Architecture (PISA) is not efficient enough in matrix operations. The key reason is that the computing engine of Very Long Instruction Word (VLIW) in PISA architecture is not efficient when dealing with large-scale parallel homogeneous computing tasks. Focusing on this problem, this paper proposed a general computing in-network computing model CAInNet for AI acceleration. Based on the traditional programmable data plane, the model innovatively integrates SIMD (Single Instruction Multiple Data) and MIMD (Multiple Instruction Multiple Data) computing modes, so that network devices can not only support protocol-independent network packet processing, but also perform in-network computing on the data carrying AI inferring and training during packet transmission. In order to verify the capability and effect of CAInNet in in-network computing and network programmability, we use INT (In-band Network Telemetry) to realize network visualization, and deploy MLP (Multi-layer Perceptron) model to realize AI-based packet classification instead of traditional TCAM table look-up routing method. Experiments show that the packet classification method based on machine learning inference can achieve 98.3% accuracy under 5K size routing table entries, and save 98.7% storage space, which can solve the routing explosion problem well. Compared with the existing methods, deploying machine learning inference in CAInNet does not increase the processing latency of programmable data plane, and only consumes a moderate amount of computing resources. This paper provides ideas and solutions for how to map neural network models in programmable data plane pipelines. In the next step, we will extend PHV to support a larger number of parameters and map more complex neural network models (e.g., convolutional and recurrent neural networks) in the programmable data plane to support more complex application scenarios.

Keywords AI hardware acceleration; communication and computing integration; in-network computing; programmable network; packet classification; deep neural networks

1 引 言

分布式机器学习模型的运行和服务提供离不开算力和网络支持。随着摩尔定律逐渐放缓,算力增长速度远低于 I/O 速度,近数据处理已经成为后摩尔时代必然的选择,简而言之,就是尽可能减少数据搬移,让数据在传输路径上得到处理。

高速网络作为数据的传输路径,将多个计算设备连接在一起,形成一个互相通信和协作的系统。在 AI 应用中,高速网络是连接着算法、数据和算力的桥梁。它使得不同计算设备之间可以共享和传输数据,实现数据的集中存储和分布式处理。随着高速网络所承载的业务流量规模和类别持续增大,近年来,研究人员使网络也具备了分布式计算的能力,多台计算和网络设备可以协同工作,加速算法的训练和推理过程。高速网络的发展和创新不仅扩展了 AI 的规模和能力,也为 AI 应用的部署和交互提供了更多可能性。

数据平面可编程技术通过可重构的匹配转发模型与定制指令的专用网络处理器实现了在不替换物理设备与网络处理芯片的情况下支持新型网络协议与网络应用。RMT[1](Reconfigurable Match-Table Architecture)架构与用于对 RMT 进行编程控制的 P4 语言由于灵活性与性能方面的优势成为网络设备和网络服务提供商的关注热点。RMT 在 Open-

Flow[2]基础上，通过协议无关的报文头解析器，支持超长指令字（Very Long Instruction Word，VLIW）的动作执行引擎等功能模块实现了可编程性和处理性能的较好统一。

但目前 RMT 架构与用于在 RMT 架构下描述网络转发行为的 P4 语言仍然处在早期发展阶段，在有状态报文处理、复杂计算（如 AI 推理）、多租户隔离与资源共享等方面存在着明显不足。例如，Gebara 等人[3]指出，RMT 架构仅可利用有限的片上内存管理简单状态，目前难以支持有状态网络报文处理，这将导致涉及大量状态信息的网内计算难以部署。

AI 分布式训练中每个节点独立进行前向传播和反向传播，然后聚合更新参数，以实现模型的同步更新，从而加速深度学习模型的训练过程，网内 AI 分布式训练是指将参数聚合过程放置在更靠近数据的位置，也就是在网络中，能够有效降低分布式计算产生的大量通信开销，使得数据在网络传输过程中同步完成参数的聚合，并不引入显著的延迟与吞吐开销[4-7]。从在网 AI 分布式训练需求来看，随着训练参数量的激增，将一部分分布式应用的计算任务卸载到高速网络的网卡或交换机能够潜在提升分布式应用的性能表现，并发挥网络的关键作用。

而从在网 AI 推理需求来看，当前面向 P4 语言的可编程网络除了使得网络协议更加软件定义外，还使得网络能够借助高能效比的算力为分布式应用提供网内计算加速服务。然而，当前典型的可编程数据平面（Programmable Data Plane，PDP）架构如协议无关交换架构（Protocol Independent Switch Architecture，PISA）在进行矩阵运算等方面还表现得不够高效。我们分析该缺陷的关键原因在于：PISA 架构中基于超长指令字的多指令多数据流（Multiple Instruction Multiple Data，MIMD）模式不适合需要大量计算且所有计算单元执行相同操作的情况，相比于单指令多数据流（Single Instruction Multiple Data，SIMD）模式成本高且速度慢。例如采用 MIMD 进行 n 元素的向量与 $n\times n$ 矩阵做矩阵向量乘需要 n^2 条乘法指令以及 $n\times(n-1)$ 条加法指令，且超长指令字中所需的 Crossbar 造成了大量不必要的硬件资源开销，与单阶段指令数的平方成正比。由于每个执行单元的指令流都是相同的，SIMD 模式将指令的获取时间均摊到每一个执行单元。而 MIMD 模式的设计主要是为了处理不同指令流，当指令流出现分支，它不需要对线程进行阻塞。然而它需要更多指令存储以及译码单元，这就意味着消耗更多的硬件资源，同时，为了维持多个单独的指令序列，它对指令带宽的需求也非常高。一般使用 SIMD 与 MIMD 的混合模式才是最好的方案。用 MIMD 的模式处理控制流，用 SIMD 的模式处理大数据，在 CPU 上使用 SSE/MME/AVX 指令扩展集就是采用的 SIMD 与 MIMD 的混合模式，而在 GPU 上，当线程束与线程块以高粒度处理分支情况时也采用的混合模式。

另一方面，随着网络设备的日益增加，核心网的路由表日益膨胀，路由表条目迅速增加到无法有效处理的程度，从而导致网络性能下降、资源浪费、路由不稳定等问题，即“路由爆炸”。传统的基于查表的路由方法需要存储大量的路由表项，而采用机器学习推理替代路由查表将节省大量的存储空间，原因是部署机器学习模型所消耗的内存资源远远小于维护路由表所需要的。因此我们在可编程数据平面中设计实现机器学习推理代替路由查表过程。

本文提出了一种面向 AI 加速的通算一体网内计算模型 CAInNet，CAInNet 是一种软硬件融合框架，在软件层面上，其包含 CAInNet Runtime 实时管理工具以及融合支持 P4 编程语言的编译器。该架构基于 RISC-V 指令，在支持所有 P4 可编程数据平面能力的基础上，融合了 SIMD 与 MIMD 两种计算模式，形成 ALU 与 AI 计算单元并行计算架构，使得交换机与网卡不仅能够支持协议无关网络分组处理，还能在数据传输过程中对承载 AI 推理与训练的分组数据实现网内计算加速。本文以解决路由爆炸问题为例验证了该模型在报文分类中推理的准确度以及性能，并为网内遥测、入侵检测以及新型协议处理提供一种更加灵活敏捷的高性能处理平台。

为了验证 CAInNet 在网内计算以及网络可编程方面的能力和效果，在该模型中使用带内网络遥测实现网络可视化，并部署多层感知机（Multilayer Perceptron，MLP）模型实现基于 AI 的报文分类，替代传统的基于 TCAM 查表的路由方法，解决核心路由表的“路由爆炸”问题。

与已有工作相比，本文贡献如下：

（1）本文提出了一种面向 AI 加速的通算一体网内计算模型 CAInNet，通过在经典 RMT 架构基础上扩展基于 SIMD 的机器学习推理的计算单元，将神经网络模型映射在可编程数据平面中，同时满足 AI 分布式训练与网内实时推理需求；

（2）在软件层面上设计 CAInNet Runtime 实时管理工具以及融合支持 P4 编程语言的编译器，使用户可通过 P4c 语言对 CAInNet 的具体网络行为进行编程，并采用与 CAInNet 后端兼容的 P4c 编译器将对应的表结构逻辑与表项内容配置到 FPGA

端的 CAInNet 流水线进行实际系统验证与测试；

(3)本文通过基于机器学习推理的报文分类案例验证了该架构的可行性，实验表明，采用机器学习推理的报文分类方法在包含 5k 路由表项的场景下准确度能够达到 98.3%，节省了 98.7%的存储空间，能够很好地解决路由爆炸问题。

2 相关工作

2.1 网内计算

大型 AI 模型由于其模型层次多，参数规模大，几乎不可能仅靠单计算节点算力完成模型训练。目前业界广泛采用的方法是通过模型并行、数据并行或者两者相结合的方式对大型 AI 模型进行训练。分布式大型 AI 模型训练的进行需要算力和网络支持。随着摩尔定律逐渐放缓，I/O 设备的速率越来越高，近数据处理已经成为后摩尔时代必然的选择。而网内计算就是分布式场景下近数据处理的一种典型方法。总而言之，就是尽可能减少数据搬移，让数据在传输路径上得到处理。网内计算指的是将一部分数据处理能力卸载到网络中，接近数据源和终端设备的地方，以实现低延迟、高效率的计算。

基于 P4[8]或微码编程的可编程交换机以及发展于 SmartNIC 的 DPU(Data Processing Unit)，实际上都是网内计算技术的具体落地实现。网内计算的意义至少有以下三个方面：首先，从应用角度分析，由于网络通信量减少，同时某些计算任务的效率得到提高，应用的整体性能得到了一定的提升，典型的应用场景包括网内键值存储、分布式数据库网内聚合以及分布式训练网内聚合等；其次，从资源利用率方面去考量，处理器的计算任务被分担，并且延迟被进一步降低，这对于数据中心多租户的环境有较多好处，典型应用包括：基于智能网卡的 Service Mesh 卸载、传输层协议卸载、加解密引擎加速等；最后，从网络本身角度出发，整体网络流量减少，网络拥塞以及其所造成的排队延迟与丢包问题能够得到一定的缓解。

从网内计算端节点硬件架构来看，可以支持 AI 硬件加速的有四个可选方案，它们分别是 CPU、GPU、FPGA 和 ASIC，如果对这几类器件的特性进行比较，会发现按照从左到右的顺序，器件的灵活性/适应性是递减的，而处理能力和性能功耗比则是递增的。

CPU 是基于冯·诺依曼架构，虽然很灵活，但由于存储器访问往往要耗费几个时钟周期才能执行一个简单的任务，延迟会很长，应对神经网络(Neural Network，NN)这种计算密集型的任务，功耗也会比较大，显然最不适合做 AI 加速。

GPU 具有强大的数据并行处理能力，在做海量数据训练方面优势明显，如 NVIDIA A100 加速卡。而推理计算通常一次只对一个输入项进行处理，GPU 并行计算的优势发挥不出来，再加上其功耗相对较大，所以在 AI 推理方面也不是最优选择。

从高性能和低功耗的角度来看，定制的 ASIC 似乎是一种理想的解决方案，但其开发周期长、费用高，对于总是处于快速演进和迭代中的机器学习算法来说，灵活性严重受限，风险太大，在 AI 加速中人们通常不会考虑它。

硬件可编程的特性使 FPGA 能够对机器学习的需要做针对性的优化，提供充足的算力，而同时又保持了足够的灵活性。如今基于 FPGA 的 SoC 平台，除了可编程逻辑，还会集成多个 ARM 处理器内核、DSP、片上存储器等资源，机器学习所需的处理能力可以很好地映射到这些 FPGA 资源上，而且所有这些资源都可以并行工作，即每个时钟周期可触发多达数百万个同时的操作，十分适合 AI 计算加速。如 IIsy[9]架构在 NetFPGA SUME[10]开发板上加速机器学习算法进行报文分类，实现了真实场景中的全线速率分类。同时，当前的可编程数据平面则能够为网内计算提供强大的算力以及灵活性。

2.2 可编程数据平面

网内计算需要两个先决条件：第一，它要求交换设备具备一定的网内基础计算和数据存储能力以支持对参数的更新；第二，在可编程的同时还需要能够对数据包保持线速处理。而当前可编程芯片就可以满足上述条件。当前可编程交换机的数据转发能力已经达到 64×100 Gbps 以上。因此采用可编程交换机进行网内计算成为了解决机器学习通信瓶颈的新思路。

PISA 作为当前典型的可编程数据平面架构采用流水线的处理方法可重构地处理网络中的数据包。英特尔 Tofino[11]作为世界上首款最终用户可编程以太网交换机就是采用 PISA 构建。如图 1 所示，可重构匹配表(Reconfigurable Match Table，RMT)作为典型的 PISA 架构，确定了基本的最小动作原语集，以指定报头在硬件中如何处理，它允许在不修改硬件的情况下通过修改匹配字段更改转发平面，相比于 OpenFlow 允许更全面地修改所有报头字段，RMT 主要包括三个部分：

(1)可编程解析器。用于解析数据包，将数据包中的关键信息与中间信息提取进报文头向量(Packet

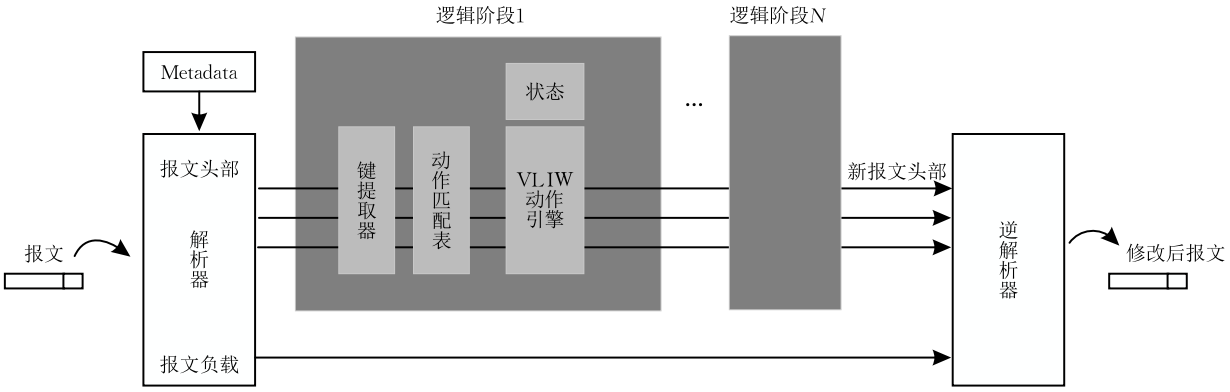

图 1　RMT 架构[1]

Header Vector，PHV），以供后续的匹配-动作流水线进行计算和修改。该过程中提取字段的位置、长度均可由用户编程，实现可编程的解析。

（2）匹配-动作流水线。PHV 进入匹配-动作流水线后，每个阶段提取关键字并匹配相应的超长指令字，由动作引擎根据 VLIW 完成对 PHV 的计算和修改。

（3）可编程逆解析器。将修改后的 PHV 写回数据包的相应位置，完成对数据包的处理并发送。

这三部分均为可编程模块，用户能够根据自身的需求对解析和处理流程进行重构，以改变流水线处理数据包的方式，灵活实现不同的网络协议处理。匹配-动作流水线能够显著降低数据包的处理延迟，有利于数据包的线速转发，且功耗低于当前传统交换机，可编程特性能够使其灵活支持各种不同的计算需求。

Trio[12]是一种用于瞻博（Juniper）网络 MX 系列路由器和交换机的可编程芯片组。如图 2 所示，Trio 的架构基于一个多线程的可编程数据包处理引擎和一个分层的大容量内存系统，这使得它与基于流水线的架构有着根本的不同。Trio 以 RTC（Run To Completion）的方式处理各种网络用例和协议，使其成为新兴网络内应用的理想平台。尽管 Tofino 交换机可以进行线速数据包处理，但其流水线结构的可编程性较为有限，而 Trio 采用处理器能够更加灵活地处理各种计算任务，其次，Trio 的大内存和对数据包尾部数据的快速访问使网络内的高效计算成为可能，Trio 的共享内存系统提供了几 GB 的存储空间，在多个应用程序同时运行的情况下，这也足够用于数据存储，最后 Trio 对单个数据包的指令数量没有限制，能够启动大型数据包所需的计算指令，可更好地用于网内计算。

2.3　PDP 加速网内计算

在参数聚合方面，SwitchML[5]是由 Barefoot 公司主导提出的交换机 AI 分布式训练加速系统。

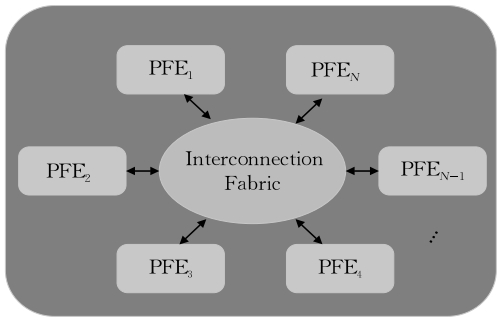

图 2　Trio 架构[12]

其主要设计思想是使用可编程交换机替代机器学习中传统的参数服务器，利用交换机的高吞吐率来加速参数更新。由于交换机的存储容量仅仅只有数十兆字节，不足以完全容纳需要的全部参数，因此，SwitchML 设计了一种分块更新的方法，即节点每次只发送一小部分参数到交换机实现同步，待收到更新结果后再进行下一个分块的更新。通过这种方式来解决交换机存储空间不足的问题。结果表明，SwitchML 对于通信量较大的神经网络结构具有显著的加速效果，加速比最大可达 3.0；而对于通信量较小的网络，也有一定的效果，加速比最小为 1.2。OmniReduce[4]利用 Tofino 可编程交换机实现了一个高效的网内聚合系统，利用稀疏性，通过只发送非零数据块来最大化有效的带宽使用，将分布式训练的速度提高了 8.2 倍。即使在 100 Gbps，OmniReduce 为遇到网络瓶颈的深度神经网络（Deep Neural Network，DNN）带来 1.4～2.9 倍的性能提升。

虽然使用可编程交换机简化了对新型网络功能与网络协议的部署，但有两个重要的限制。首先，当前的可编程交换机只支持定点算术。因此，SwitchML 需要将特定模型的浮点梯度转换为定点表示，这可能会影响达到目标精度所需的训练时间。其次，Tofino 交换机在不同的流水线中不保持状态，并且每个流水线中的阶段数量有限。

目前在学术界,研究人员也探索了利用可编程网络处理技术在网内进行 AI 推理加速的可行性。在神经网络推理方面,Taurus[13]是一个用于线速推理的数据平面。Taurus 将灵活的、抽象了并行模式(MapReduce)的定制化硬件添加到可编程网络设备(如交换机和网卡)中,这种新硬件基于流水线 SIMD 实现并行,完成逐包 MapReduce 操作(如推理)。对 Taurus 交换机 ASIC 的评估表明,Taurus 的运行速度比基于服务器的控制平面快几个数量级,同时面积增加了 3.8%,线速机器学习模型的延迟增加了 221 ns。此外,Taurus FPGA 原型实现了完整的模型精度,比最先进的控制平面异常检测系统的能力提升了两个数量级。然而,Taurus 将神经网络计算单元放置在两个可编程流水线之间,无法完成与普通数据包的协同处理,对需要进行神经网络推理的数据包来说增大了处理延迟。

在非神经网络推理方面,IIsy[9]将训练有素的机器学习模型映射到匹配-动作流水线,探索了商用可编程交换机用于网络内分类的潜在用途。IIsy 是一个软硬件结合的系统,讨论了映射到不同目标的适用性,如决策树、朴素贝叶斯、K-means 聚类以及支持向量机 SVM,该解决方案可以推广到其他非神经网络机器学习算法。然而该方法仅讨论了非神经网络在匹配-动作流水线中的实现方法,且仅采用查表的方式完成,以损失推理精确度为代价,并对流水线中每阶段的存储能力提出了较高的要求。Henna[14]是一种分层分类系统的第一个交换机实现,提出了一种与 PISA 架构内部组织一致的设计,并集成了将决策树映射到交换机硬件的策略。使用 P4 语言将 Henna 实现为使用现成的 Intel Tofino 可编程交换机的真实测试平台。实验表明,Henna 将高级单阶段模型的 $F1$ 分数提高 21%,同时将交换机资源的使用率平均保持在 8%。然而 Henna 采用的决策树组织架构仍要存储规则信息,相对于神经网络中只存储权重的做法来看并未减小规则集的存储开销。考虑到决策树的实现消耗了大量的交换机资源(如阶段和内存),且更复杂的网络学习模型的高计算复杂度和大存储需求给在交换机上的部署带来了挑战。Mousika[15]作为一种网络智能框架,将决策树修改为二叉决策树。支持更快的训练,生成更少的规则,更好地满足交换机约束。其次,Mousika 引入了师生知识蒸馏,实现了从其他学习模型到二叉决策树的通用转化。通过转化,不仅可以利用复杂模型的超强学习能力,而且可以避免直接部署在交换机上进行线速处理时的计算与存储限制。对于流量预测任务,蒸馏二叉决策树将决策树的准确率提高了约 3%,规则数量比普通二叉决策树减少了约 80%。然而,Mousika 仅仅在流量大小预测、流量分类和恶意软件检测场景中进行了实验,本质上还是将决策树模型映射在可编程数据平面中,所需存储空间仍随规则集的增大而增大,难以支持大规模规则集,如大型路由表。NetBeacon[16]引入了多阶段顺序模型架构,用于分析流不同阶段的数据包,并结合流级功能以实现高学习精度。针对在网络数据平面上部署树状模型时出现表条目爆炸问题,提出了一种高效的模型表示机制,改进了可扩展性,以便通过多个紧密耦合的设计来处理并发流,以管理有状态存储。NetBeacon 在流量分析准确性和硬件表消耗方面优于 Mousika 等现有方法。但是,NetBeacon 采用的决策树模型与 MLP 相比,对于非线性计算的支持不够,只能进行准确度有限的分类问题,故 NetBeacon 的应用领域也设置在检测分布式拒绝服务攻击等对准确度不敏感的安全应用中。

2.4 基于机器学习的报文分类方法

NuevoMatch[17]采用范围查询递归模型索引(Range Query Recursive Model Index,RQ-RMI),显著压缩规则集索引,使其完全适合 CPU 缓存(L1/L2),并将神经网络采用决策树的形式组织起来,在内存占用、延迟和吞吐量方面优于现有技术,可以将具有多达 500 K 规则的规则集压缩到现代处理器的小型缓存中。最大的规则集(500 K)中,与 Cutsplit、NeuroCuts 和 TupleMerge 相比,NuevoMatch 并行实现的延迟减少了 62.96%、77.27%和 46.17%,吞吐量提高了 1.3 倍、2.2 倍和 1.2 倍。对于具有 100 K 规则的分类器,增益较低,但仍然显著:延迟分别减少了 50%、72.22%和 61.54%,吞吐量分别高 1.0 倍、1.7 倍和 1.2 倍。虽然 NuevoMatch 降低了存储规则集的存储开销,但由于其采用C++进行推理计算,仍然占用 CPU 的计算资源,且将多个神经网络组织成决策树的形式将耗费大量的训练时间。

报文分类是服务质量、网络安全等许多网络服务的关键组成部分。这些网络服务要求报文分类速度尽可能快,使用更少的内存并支持可扩展性。此外,软件定义网络交换机规则集的高维性和大规模性给报文分类带来了新的挑战。MBitTree[18]主要改进了现有的决策树算法。首先,引入一种新的规则集划分技术,实现规则集的自适应快速划分;其次,采用一种新的多比特裁剪方案,在引起规则复制较少的情况下,构建较短的树;MBitTree 具有较高的分类速度和良好的可扩展性。实验结果表明,与

CutSplit 相比，MBitTree 最多减少了 85.69% 的内存消耗，最多减少了 41.18% 的内存访问次数。在 NetFPGA 上，对于 10 K 大小的规则集，MBitTree 可以达到 100 Gbps 以上的吞吐率。

软件定义网络和云计算需要频繁地更新规则集以灵活地进行策略配置。元组空间搜索（Tuple Space Search，TSS）[19] 采用 Open vSwitch（OVS）实现。在 TSS 中，每个元组由一个哈希表管理，对数据包进行分类需要遍历所有的哈希表。合并元组可以减少哈希表的数量，但不可避免地会增加哈希冲突，在某些情况下甚至会恶化分类性能。现有算法不能同时满足快速报文分类和在线规则更新的要求。CRP[20] 是一种基于卷积神经网络（Convolutional Neural Network，CNN）的范围划分方法，以实现快速的报文分类和在线更新。CRP 利用基于 CNN 的图像识别技术，在规则集分布发生变化时，快速将元组划分到范围空间中，减少了哈希操作，同时避免了将大量规则映射到哈希表同一位置所造成的规则重叠。实验结果表明，与同类算法相比，CRP 算法的分类速度平均高 3.2 倍，更新速度平均高 4.2 倍。

2.5　基于神经网络推理的路由决策方法

在可编程网络中，使用神经网络推理进行路由决策正呈现出巨大的发展趋势。Martín 等人[21] 实现了机器学习路由计算（MLRC）模块，利用 ONOS 控制器构建了一个名为 MLRC 的机器学习模型，用于寻找 SDN 网络中不同路径的优化。由于其简单性和可解释性，MLRC 实现了一个逻辑回归分类器。根据结果显示，SDN 网络能够重新计算其路由配置，并在非常短的时间内执行，处理流量矩阵中的输入变化。

RoPE 由 Sacco 等人[22] 提出，是一种基于预测带宽来适应底层边缘网络路由策略的架构。RoPE 是一组监督时间序列模型和机器学习方法的集合，通过训练来预测带宽，控制器可以检查应用程序是否适合网络负载。它自动选择应用的算法，以保证最佳性能。为给定的用例选择正确的预测方法是许多因素的函数，例如可用的历史数据和外部变量。训练数据是通过 Mininet 模拟器收集的。SDN 控制器会跟踪过去的链路负载情况，如果预测当前路径将发生拥塞，则会选择一条新的路由。

Sun 等人[23] 结合多种机器学习（Machine Learning，ML）算法提出了一种称为 MACCA2-RF&RF 的数据流分类方法，以高精度分类数据流并获得 QoS 需求。作者使用真实数据集和基于 Floodlight 和 Mininet 的 SDN 实现进行了全面评估，非常接近真实场景。然而，该方法也存在不足，例如安装的表项的数量应该减少以提高可扩展性。

EL-Garoui 等人[24] 利用 SDN 和 ML 在智慧城市中实现高效路由，其中大多数应用程序都基于物联网。他们开发了一个基于朴素贝叶斯算法的框架，并基于蒙特利尔城市开放数据网站和 SUMO 城市移动模拟器创建了一个数据集，在时延和数据包传输率方面取得了较好的效果。

Owusu 等人[25] 提出了不同的 ML 模型实现，对 SDN-IoT 网络中的流量进行分类，以进行流量工程。比较了三种不同的分类器：随机森林分类器、决策树分类器和 K-means 分类器。根据他们的分析，采用随机森林分类器获得了最佳准确率 0.83。

3　CAInNet 网内计算架构设计

CAInNet 首先要满足 RMT 架构的核心需求，这要求 CAInNet 拥有可编程协议无关解析、通用关键字查表以及通用动作指令执行等功能；同时，作为一个可编程数据平面，需具备与生产场景相适配的线速的报文转发能力，才能具备真正的科研价值及工业价值；在此基础上要能够支持网内计算中的机器学习推理。

CAInNet 是一种软硬件融合架构，在软件层面上，其包含 CAInNet Runtime 实时管理工具以及融合支持 P4 编程语言的编译器。在硬件层面上包含安全预处理过滤器、可编程解析器、关键字提取器、自定义匹配查表引擎、动作执行引擎、AI 计算单元、参数配置模块、逆解析器等八个模块，采用 Virtex-7 690T FPGA 实现，融合 SIMD 与 MIMD 两种计算模式，SIMD 在获取数据和执行指令的时候，都做到了并行，对于那些在计算层面存在大量数据并行的计算中，使用 SIMD 是一个很划算的办法，例如实践中的向量运算或者矩阵运算。因为在处理向量计算的情况下，同一个向量的不同维度之间的计算是相互独立的。因此，我们采用这种“数据并行”加速方案，融合 SIMD 与 MIMD 两种计算模式，矩阵向量乘操作仅需一条指令。总的来说，CAInNet 不仅支持协议无关网络分组处理，还可对承载神经网络推理的分组实现网内计算加速，且不引入额外的处理延迟。带内网络遥测负责收集网络数据上传控制平面进行训练，实时监测网络状态。

3.1　CAInNet 数据平面实现

CAInNet 需要支持 DNN 中的推理计算，大部分神经元中的推理过程是采用矩阵向量乘法操作完成的，因此需要在该架构中集成 AI 计算单元以支

持矩阵向量乘法操作。

3.1.1 多层感知机模型

感知机作为第一个人工神经网络，它的意义重大。但是它的缺点也是特别明显，网络过于简单，不能解决非线性问题等。为此，人们又提出了一种新型的感知机——多层感知机[26]。MLP 是在单层神经网络基础上引入一个或多个隐藏层，使神经网络有多个网络层。隐藏层位于输入层和输出层之间。图 3 展示了一个 MLP 的神经网络图。

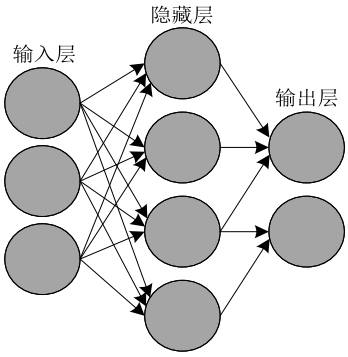

图 3 多层感知机示意图

MLP 作为便于硬件实现且通用的神经网络模型十分适合映射在 RMT 流水线结构中。受 SIMD 和 MLP 的启发，考虑到流水线架构对数据包的流式处理，在流水线各阶段之间单独放置 AI 计算单元将提高流水线的处理延迟，因此可将每个 AI 计算单元分散在各个流水线阶段内，能够隐藏矩阵向量乘法带来的延迟，并可根据编程选择是否进行矩阵向量运算。每个 AI 计算单元完成一层神经元的计算功能，在某一阶段得到推理结果后，后续阶段可利用该结果进行相应的决策和处理。AI 计算单元与动作引擎内的 ALU 并行执行且互不影响，既保留了流水线流式处理的高性能，又增加了可选择的机器学习推理能力。

3.1.2 AI 计算单元设计

为了满足上述的设计需求，CAInNet 结构主要包括安全预处理过滤器、可编程解析器、关键字提取单元、自定义匹配查表引擎、指令执行引擎、参数配置模块、AI 计算单元以及可编程逆解析器关键子模块。CAInNet 的报文处理流程如图 4 所示。

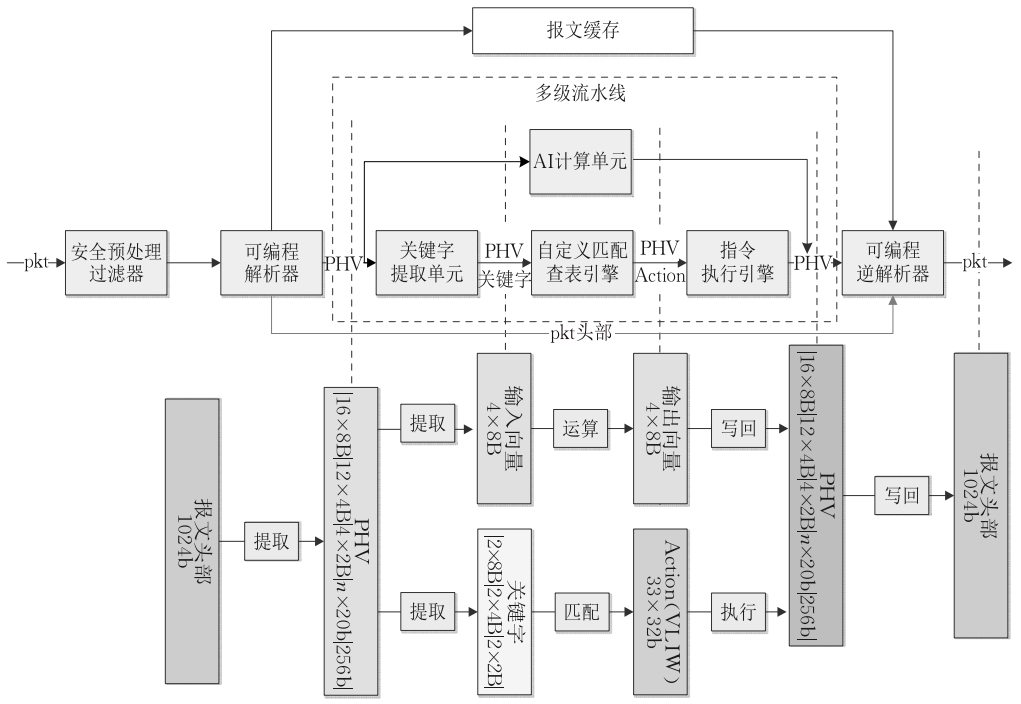

图 4 CAInNet 报文处理流程

为了使 CAInNet 支持 AI 推理加速，我们在该架构中添加了 AI 计算单元与权重参数配置模块，AI 计算单元分散在各个流水线阶段中，与 ALU 并行处理 PHV，权重配置模块负责接收控制报文中携带的矩阵参数并配置给相应阶段的 AI 计算单元。每一个 AI 计算单元完成 MLP 中一层神经元的操作（即矩阵向量乘、增加偏置值以及激活函数），整条流水线能够完成一次神经网络的推理过程，CAInNet 的总体架构如图 5 所示。

AI 计算单元（AICU）包含矩阵向量乘、偏置值以及激活函数模块，如图 6 所示。AICU 对传入的 PHV 判断标志位（如 VLAN ID），若需要进行矩阵

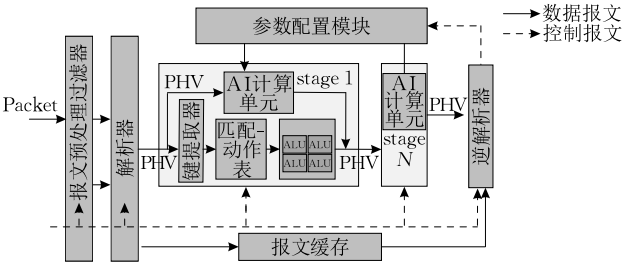

图 5　CAInNet 总体架构

运算，则提取前 4 个 8B 容器的数据，构成 1×32 大小的向量，每个向量元素大小为 1B。对于数据长度达不到 32 的向量，后面补 0，不影响矩阵向量乘法结果。将该向量发送给每级 32 个乘加器(MA)，每个乘加器中放置1×32的权重列向量，执行一次向量点乘运算，用不到的全部置 0，不足 32 的也在后面补 0。即每个 AICU 实际执行 $M_{1\times32}N_{32\times32}$ 的矩阵向量乘，通过置 0 可以实现不同大小的矩阵向量乘，以此进行升维或降维。例如要完成 $M_{1\times16}N_{16\times16}$ 的矩阵向量乘法操作，则将输入向量的后 16 个元素置 0，送入 32 个乘加器，其中后 16 个乘加器的权重参数全部置 0，因此得到的运算结果的前 16B 为实际结果。每个阶段输出一个 32 元素的向量，与输入的 PHV 位置相对应写回。最后一个神经元得到1B的结果。Deparser 将结果写入 metadata。若报文不需要进行神经网络推理，则前 4 个 8B 容器的值仍然参与 ALU 中的运算并写回 PHV。

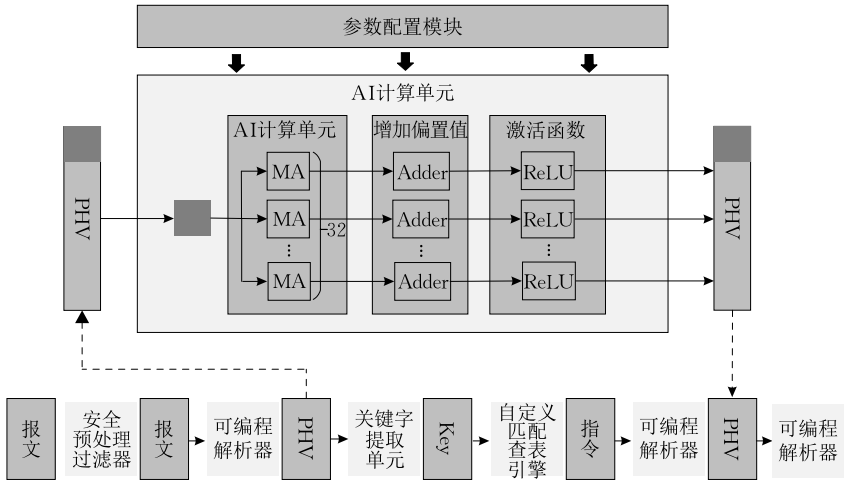

图 6　AI 计算单元示意图

矩阵向量乘模块的设计如图 7 所示，每个乘加器共包含 6 个流水线阶段，第 1 阶段完成向量中每个元素的乘法，后续阶段由加法树完成 32 个乘积的加法操作。该方法采用空间换时间的方法利用乘加器的并行计算将向量复制多份同时进行点乘运算，完成矩阵向量乘法，且运算总延迟不超过 ALU 处理的延迟，两者互不影响，既能够选择性实现机器学习运算又不降低原有流水线的性能。

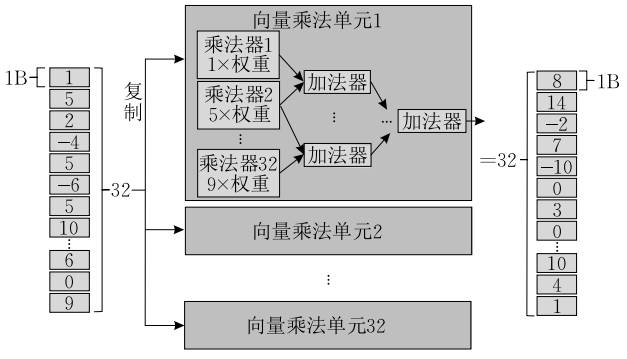

图 7　向量乘加器示意图

3.1.3　权重参数配置模块

权重参数配置模块负责接收控制报文中携带的所有神经元参数，包括矩阵参数以及偏置值，并将其分配给对应阶段中的 AI 计算单元，控制报文中的参数格式如图 8 所示。

配置模块独立于每个 stage，负责配置所有 stage 中 AICU 的权重参数。控制通路从 Deparser 连接到配置模块。用控制报文中的 Module ID 指示配置模块。控制报文中高 32 个 32B 为矩阵参数，低 1 个 32B 为偏置值参数，配置 N 个阶段需要 N 组参数。

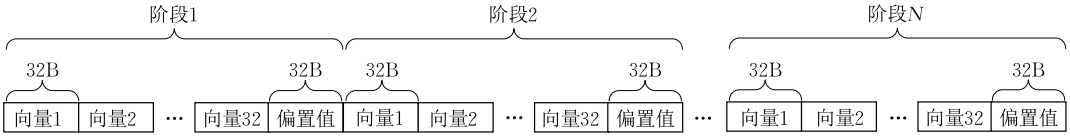

图 8　参数配置格式

3.1.4 多租户隔离设计

现有 RMT 架构仅可将报文头部提取的字段以及架构自定义的 metadata 字段用于匹配动作，无法对节点内突发事件进行响应。而随着更多网络服务（如负载均衡器、防火墙、网内遥测等）和网内计算应用（如键值存取、分布式机器学习等）被部署到云数据中心交换机与智能网卡侧，基于 RMT 架构的转发设备就应当支持不同网络业务在单节点的逻辑隔离与资源共享，从而满足安全与性能等多方面要求。

CAInNet 的可编程协议无关解析单元主要根据用户配置的系统参数对不同协议的报文信息进行解析和提取，将报文头信息、元数据和比较指令等信息转换为报文头向量（Packet Header Vector，PHV）。为了简化过程，在解析过程中将 PHV 统一成结构相同的格式。同时支持根据 VLAN ID 或 VNI 进行面向不同租户虚拟网络的隔离解析。对于任何报文，可编程解析器通过 VLAN ID 或 VNI 作为索引查询解析器内部基于 FPGA 的 BRAM 单元实现的协议解析表得到一条解析表项。用于并行提取 PHV 的填充字段。每条解析表项包含提取字段相较于报文头起始偏移、字段长度类型、字段放置于 PHV 中的索引（与字段长度类型共同可以确定一个字段在 PHV 的位置）。最低位为有效位，用于标识当前子指令是否有效。关键字提取模块用于从 PHV 中提取匹配关键字，并且同样支持根据 VLAN ID 或 VNI 进行面向不同租户虚拟网络的关键字提取。该模块包含一个基于 BRAM 单元的关键字提取规则查找表。查找表的表项索引为随 PHV 同步传输的 VLAN ID 或 VNI 字段。动作执行模块内存储资源被分为若干区域，各区域根据用户 ID 信息生成不同的用户空间，并生成以用户 ID 为索引，区域起始地址和用户空间长度为表项的索引表，实现各用户之间资源隔离，以支持多租户场景，确保用户空间安全。

3.2 CAInNet 控制平面实现

我们设计了一个无状态的方法来修改流水线中的表项，即使用一组专门的报文（即控制报文）来修改表项。控制报文可以从软件端生成，里面包含需要修改的表号和修改的内容。在报文头字段中，用专门的字段来标识该报文针对的模块号，表项的内容包含在有效负载字段中。报文被流水线接收后，它将被每个流水线的模块识别，并一路通过流水线。如图 5 所示，每个模块将检查该模块是否是报文的目标，从而控制模块是读取有效负载并相应地修改表项，还是把报文传递给下一个模块。如果目标都不匹配，则在报文出流水线之前丢弃控制报文。

我们对流水线中的模块使用两层索引：除了解析器、参数配置模块和逆解析器（它们在流水线中只出现一次）之外，所有其它模块（关键字提取单元、自定义匹配查表引擎和指令执行引擎）都用 8b 的模块号（Module ID）来表示。较高的 5b 标记它所属阶段，较低的 3b 区分它是关键字提取单元、自定义匹配查表引擎还是指令执行引擎。

通过控制通路，控制报文能够进入参数配置模块，由参数配置模块统一完成对 AI 计算单元中神经网络参数的重配置。控制报文配置 AI 参数的过程如图 9 所示。控制报文进入参数配置模块后，由参数配置表决定报文内容如何分配给各个阶段的 AI 计算单元，每个阶段中参数的排列方式如图 8 所示。每个 AI 计算单元接收到神经网络参数后可快速且同步进行重配置，具备高敏捷可重构特性。

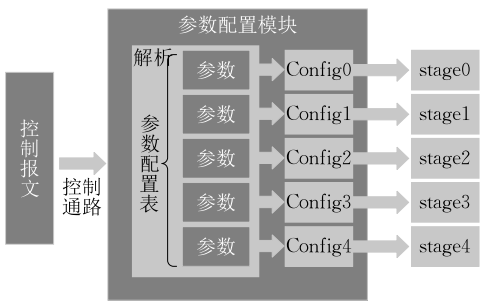

图 9 控制报文配置 AI 参数过程

为了支持对流水线中各类表项的灵活与安全配置，CAInNet 借助 UDP 报文采用带内控制方式支持流水线中各类表项的重配置。为了对数据报文与控制报文进行区分，用于表项配置的 UDP 报文的目的端口号统一采用 0xf2f1，并采用 cookie 验证机制对控制通路进行安全访问控制，以防御控制报文重放攻击等恶意行为。

3.3 语言与编译器适配

CAInNet 是一种软硬件融合架构，在软件层面上，其包含 CAInNet Runtime 实时管理工具以及融合支持 P4 编程语言的编译器。本工作基于开源 P4 编译器增加了对 CAInNet 的 P4 可编程的兼容性支持。因此，用户可通过 P4c 语言对 CAInNet 的具体网络行为进行编程，并采用与 CAInNet 后端兼容的 P4c 编译器将对应的表结构逻辑与表项内容配置到 FPGA 端的 CAInNet 流水线进行实际系统验证与测试。需要注意的是，为了简化用户编程与管理 CAInNet 的步骤，CAInNet 不仅允许用户在 P4 程序中静态描述包括协议解析表、规则匹配表在内等所有表项的内容，也支持用户在运行时采用 CAInNet

的安全可扩展控制通路对数据平面的各类表项内容进行配置。

4　应用场景分析

本文分析了 CAInNet 在 INT、报文分类以及入侵检测等场景中的应用。CAInNet 作为 RMT 架构的扩展可天然支持 INT 技术，用户能够通过自定义流水线功能选择测量对象。得到网络中的自定义测量数据后，能够通过控制平面进行模型训练，并将模型参数下发至数据平面中，实现控制平面与数据平面的灵活交互。报文分类与入侵检测都可通过 INT 在网络中收集训练数据交给控制平面进行模型训练，例如报文分类可收集目的 IP 地址及其对应的输出接口，入侵检测可收集流量状态与相应的网络异常情况，验证 CAInNet 的高度灵活性以及推理准确度。

4.1　带内网络遥测

带内网络遥测[27]是由 Barefoot、Arista、Dell、Intel 和 VMware 于 2017 年共同提出的一种不需要网络控制平面干预，网络数据平面收集和报告状态的带内网络遥测规范的后续版本。在 INT 框架中，数据平面无需控制平面的参与，能够直接收集和汇报网络状态信息。此外，INT 的测量对象常见的有：输入输出接口、队列长度、链路利用率、时延、节点 ID 和处理的报文或者字节计数等。INT 利用 P4 强大的语义表达能力优势以及 PISA 所共有的协议无关和平台无关等特点，使得它具有传统网络测量不具备的优势：可直接读取交换机内部状态，测量对象更为丰富，并且实时性好，测量粒度细。

CAInNet 作为 RMT 架构的扩展可天然支持 INT 技术，用户能够通过自定义流水线功能选择测量对象。我们所提出的 INT 系统采用 INT-MD 模式，即 INT 的指令和元数据均嵌入数据包中。在数据包进入监控系统后，INT 源节点将 UDP 目的端口修改为 INT_TBD 常量（用来指示 INT 头的存在）并且将 INT 头插入到数据包的 UDP 头部与 UDP 载荷之间。在网络中转发时，INT 中间节点收集数据平面数据并插入到 INT 预留的数据位置中。到达网络出口即 INT 宿节点时，由宿节点取出 INT-MD 头部和元数据，并将原始 UDP 的目的端口号恢复。监控系统的主机得到 INT 收集到的数据，赋能拥塞控制、故障诊断、态势感知等一系列应用场景。

INT 头部格式如图 10 所示，其中目的端口号被 INT 源节点修改为 INT_TBD，令 INT_TBD＝0Xffff，指示 UDP 头之后 INT 头的存在。INT 头中，Length 字段长 16 位，表示下一组 INT 数据插入的位置。初始值是 0，每次大小递增 5，比如第一组 INT 数据插入时，Length 的值是 0，即从 INT 头结束位置开始插入，Length＋＝5；第二组插入时 Length 为 5，故从上次插入位置向后数 5×4＝20 byte 的位置插入新收集的数据，以此类推。UDP Port 字段保存 UDP 头中原始的目的端口号。32 位 Timestamp 字段保存 INT 源节点创建 INT 头的时间。

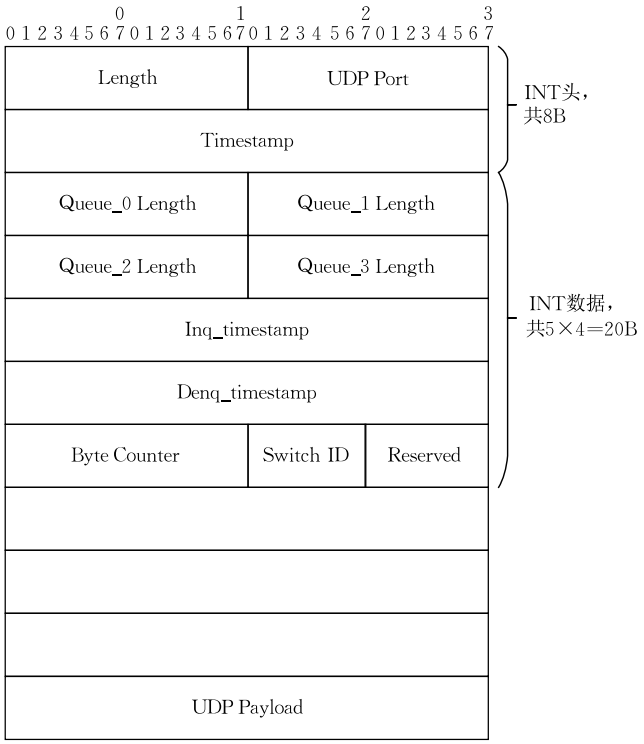

图 10　INT 头部格式

4.2　基于 AI 推理的报文分类

CAInNet 在支持所有 P4 可编程数据平面能力的基础上，融合了 SIMD 与 MIMD 两种计算模式，使得交换机与网卡不仅能够支持协议无关网络分组处理，还能在数据传输过程中对承载 AI 推理与训练的分组数据实现网内计算加速。“报文分类”是一个广义的概念，通过目的 IP 地址推理计算出转发接口本质上也是报文分类的范畴，即确定报文的类型并选择正确的输出接口。为了演示 CAInNet 在网内计算以及网络可编程方面的能力和效果，我们在该模型中使用带内网络遥测实现网络可视化，并部署多层感知机模型，替代传统的基于 TCAM 查表的路由方法，将其作为报文分类的一个应用场景，并验证其正确率与性能，解决核心路由表的“路由爆炸”问题。

本文部署于数据平面的神经网络实现了机器学

习的推理计算，而训练过程则在控制平面上进行，数据平面与控制平面通过参数配置模块互相联系，即控制平面负责训练模型，得到各层神经元参数，将需要配置的参数写入控制报文，通过控制通路发送给参数配置模块，由参数配置模块配置各个阶段中的神经元的参数寄存器，实现数据平面与控制平面的联动。

我们使用 PyTorch 开源机器学习库构建 MLP 模型，对带内网络遥测收集到的报文数据集进行训练，将 128 bit 的 IPv6 地址转换为 32 维的特征向量，每个向量元素大小为 1 B，训练中我们使用了一个输入层、两个隐藏层与一个输出层。每层神经元权重矩阵大小分别为 16×32、32×16、16×8 和 8×1。

4.3 基于网内机器学习推理的入侵检测

入侵检测[28]从计算机网络系统中的若干关键点收集信息，并分析这些信息，观察网络中是否有违反安全策略的行为和遭到袭击的迹象。网络入侵者通常利用网络的漏洞进入系统，如 TCP/IP 协议的三次握手，就给入侵者提供入侵系统的途径。任何一个网络适配器都具有收听其它数据包的功能。它首先检查每个数据包目的地址，只要符合本机地址的包就向上一层传输，这样，通过对适配器适当的配置，就可以捕获同一个子网上的所有数据包。所以，通常将入侵检测系统放置在网关或防火墙后，用来捕获所有进出的数据包，实现对所有数据包的监视。

利用 CAInNet 模型，我们能够实现网内入侵检测。通过 INT 完成网络内信息收集，在控制平面中对收集到的有关系统、网络、数据及用户活动的状态和行为等信息进行训练，构建检测模型，并将模型参数下发到 CAInNet 的 AI 计算单元中，数据平面能够在数据包传输过程中对数据包的关键字段进行推理，判断网络异常状态，记录并上传异常状态信息。

5 原型实现和实验验证

5.1 原型系统实现

我们采用 Vivado 2020.1 在 NetFPGA-SUME 上实现了 CAInNet 模型，该板卡采用 Virtex-7 690T FPGA，训练中我们使用了一个输入层、两个隐藏层与一个输出层的 MLP 模型。每层神经元权重矩阵大小分别为 16×32、32×16、16×8 和 8×1。

5.2 实验验证

5.2.1 实验设计

我们采用包含 5000 条路由表项的规则集进行 1000 轮的模型训练，每一轮中每一批次的训练输入的数据集规模大小为 128，训练损失以及最终预测精确度如图 11 所示。在训练过程中，采用交叉熵计算平均损失，每 100 个轮次打印了当前平均损失。初始轮次的损失很高（约 366129），但随着训练轮次的增加，损失逐渐减少。经过 100 轮训练后，损失下降到约 34.8，继续训练后损失进一步减小。最终，在 900 轮之后，损失可以视为足够小（约 0.298），推理准确度为 98.3%。

```
训练集样本数量：5000
测试集样本数量：5000
特征维度：16
Epoch：0001 cost= 366129.562500000
Epoch：0101 cost= 34.831417084
Epoch：0201 cost= 3.462969303
Epoch：0301 cost= 1.331723809
Epoch：0401 cost= 1.244737625
Epoch：0501 cost= 1.240414262
Epoch：0601 cost= 1.218391538
Epoch：0701 cost= 0.799913108
Epoch：0801 cost= 0.322228938
Epoch：0901 cost= 0.298148304
训练完毕！
Accuracy：0.983
```

图 11 由损失函数计算的每 100 轮损失与预测精确度

基于 CAInNet 进行 IPv6 路由推理的具体设计如下：

（1）PHV 预留前 4 个 8 B 容器进行矩阵向量运算，解析器决定是否需要矩阵运算，若需要则将 PHV 标识位 flag 字段置 1。

（2）AICU 对传入的 PHV 判断标志位，若需要进行矩阵运算，则提取前 4 个 8 B 容器的数据，构成 1×32 的向量。

（3）对于数据长度达不到 32 的向量，后面补 0 即可，不影响矩阵向量乘法结果。

（4）每级 32 个乘加器，每个乘加器中放置 1×32 的权重列向量，用不到的全部置 0，不足 32 的也在后面补 0。即每个 AICU 实际执行 $M_{1\times32}N_{32\times32}$ 的矩阵向量乘，通过置 0 可以实现不同大小的矩阵向量乘，以此进行升维或降维。

（5）针对该场景实现 4 层神经元，权重矩阵大小分别为 16×32、32×16、16×8 和 8×1。

（6）每个阶段输出一个 32 元素的向量，写回 PHV 中相对应的位置。最后一层神经元得到 1 B 的结果。Deparser 将输出接口号写入 metadata。

（7）AICU 参数由控制报文配置。一个阶段共 32×32+32=1056 B 参数。

我们将训练后的矩阵权重参数以及偏置值下发到 NetFPGA-SUME 实现的 CAInNet 模型，通过 P4 编程语言的编译器对可编程交换机进行配置，以根据 IPv6 地址推理出报文的转发接口。

5.2.2　实验平台搭建

实验环境由 3 台 3 层交换机组成，如图 12 所示。第一台交换机的 4 个网络接口分别连接主机和其余两台交换机。带内网络遥测系统收集到的队列长度、延时、链路利用率和输入输出接口数据可以帮助网络管理者实时监测拥塞情况，将网络白盒化、可视化，提高网络的管理效率。我们通过带内网络遥测获取不同目的 IP 地址数据包的输出接口，作为数据集发送给控制平面进行模型训练，并将训练参数下发给交换机，再发送报文验证输出接口的正确性。

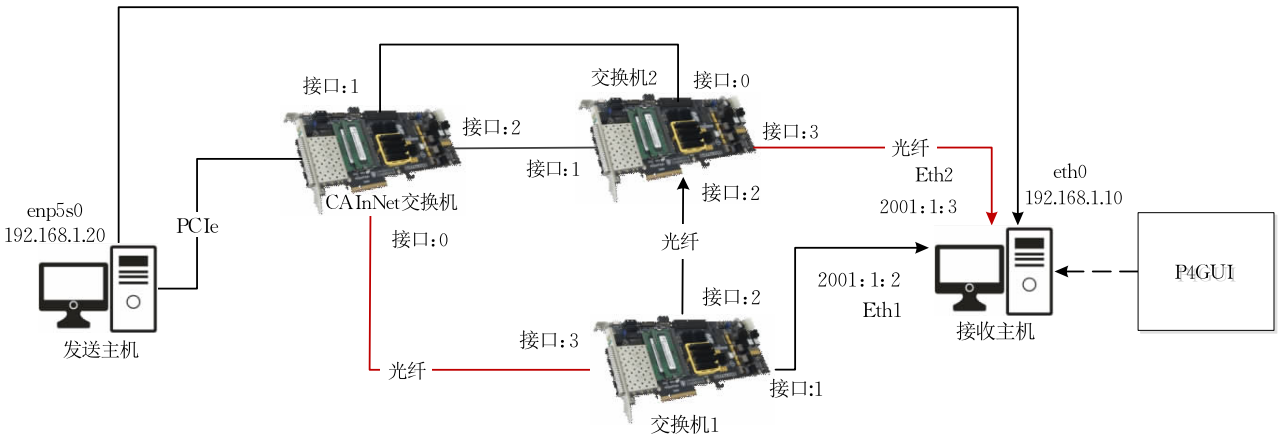

图 12　实验完整拓扑搭建

我们通过 INT 技术获得报文在交换机中的输出接口，并与原有的路由表做对比，判断发出的数据包是否按照正确的接口进行转发，并统计基于机器学习推理的路由方法的正确率。

5.2.3　实验结果

CAInNet 资源利用率如表 1 所示。由于 AI 计算单元中处理参数较多，矩阵向量乘法运算量较大，因此耗费 LUT 资源较多，但并未使用额外的 BRAM。我们使用 MoonGen[29] 生成不同大小的数据包。如表 2 所示，当数据包大小为 512 byte 时，CAInNet 可以达到 10 Gbit/s 的速率。CAInNet 延迟性能评估如图 13 所示，与 Taurus 相比，Taurus 的线速机器学习模型的延迟增加了 221 ns，而 CAInNet 由于将报文处理与神经网络推理采取了并行化处理，对于 70 B～1500 B 的报文，其延迟仅为 1 μs～1.25 μs 之间，并未在数据平面中增加机器学习的延迟。

表 1　CAInNet 资源利用率

硬件实现	Slice LUTs	Block RAMs
NetFPGA reference switch	42325(9.77%)	245.5(16.7%)
RMT on NetFPGA	200573(46.3%)	641.0(43.6%)
CAInNet on NetFPGA	398065(91.89%)	641.0(43.6%)

表 2　CAInNet 吞吐量与包转发速率

数据包大小/B	吞吐量/Gbps	包转发速率/Mpps
64	7.1	13.9
96	8.4	11.9
128	8.6	8.5
256	9.4	4.3
512	10.0	2.1

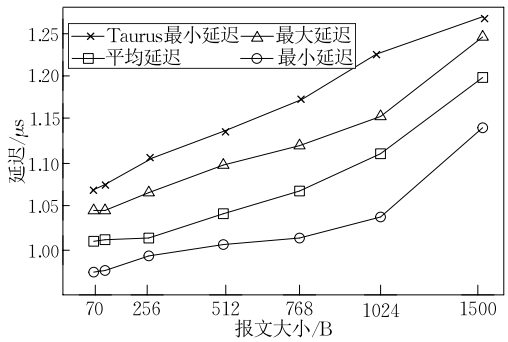

图 13　CAInNet 延迟性能评估

我们通过训练不同大小的路由表得到不同的神经网络模型，分别进行推理验证，在 5 k 大小路由表项对应的模型中能够实现 98.3% 的路由准确度，随着训练规则集的扩大，正确率也会随之提高。路由表存储一条 IPv6 地址需要 128 bit，因此传统查表方法的存储开销会随着路由表的扩大不断增加。而神经网络中的权重参数所占用的存储空间固定，并不会随着路由表项的增加而增加，相比于不同大小的路由表其节省的存储空间比例会随之增加。基于神经网络推理的路由准确率与存储优化情况如表 3 所示。

表 3　基于神经网络推理的路由准确率与存储优化

训练集大小/k	正确报文数量	路由准确率/%	存储空间优化比例/%
5	4916	98.3	98.70
50	49357	98.7	99.87
500	496488	99.3	99.99

6 总结与展望

本文提出了一种面向 AI 加速的通算一体网内计算模型 CAInNet。该模型在支持所有 P4 可编程数据平面能力的基础上,融合了 SIMD 与 MIMD 两种计算模式,克服 SIMD 在矩阵运算方面的不足,使得交换机与网卡不仅能够支持协议无关网络分组处理,还能在数据传输过程中对承载 AI 推理与训练的分组数据实现网内计算加速,形成 ALU 与 AI 计算单元协同计算的双通路架构。AI 计算单元与动作引擎中的 ALU 可并行运算且互不影响,我们在该模型中使用带内网络遥测实现网络可视化,并部署多层感知机模型实现基于 AI 的报文分类,替代传统的基于 TCAM 查表的路由方法,解决核心路由表的"路由爆炸"问题。

在网络中部署 ALU,能够在网络传输过程中根据用户自定义的运算指令进行各类运算。同时配置有状态存储器,支持 ALU 存储处理过程中的状态信息,通过加载/储存等操作,实现对携带状态的业务进行处理。

CAInNet 是一种软硬件融合框架,在软件层面上,其包含 CAInNet Runtime 实时管理工具以及融合支持 P4 编程语言的编译器,采用软件开发的方式控制数据平面的报文处理流程。目前,传统基于 FPGA 的可编程网络仅实现 P4 到 Verilog/VHDL 语言的映射,通过每次在 FPGA 上烧写以实现可编程特性。控制报文在各模块生成匹配表项,数据平面通过查表来匹配执行动作。网络的管理者通过发送控制报文,动态修改各模块匹配表,以实现在 FPGA 烧写完成后,依然可以保持动态可编程性。

对于端到端通信,通过固定指令修改、控制数据报文在各模块转发的目的接口,实现报文更加灵活高效的转发,使得网络管理者能够更加细粒度地管控每一个报文在各模块之间端到端的处理过程,实现端到端通信的可编程性。同时,减少各模块之间耦合,方便用户快速插入或修改模块,支持用户进行敏捷开发与验证。

本文为可编程数据平面流水线中如何映射神经网络模型提供了思路与解决方案。当前 CAInNet 架构中的 AI 计算单元采用 INT8 类型数据进行推理计算,对于需要浮点数计算的机器学习模型暂时还不能支持,下一步我们将通过改进 AI 计算单元的架构使其支持浮点数运算。目前我们将 PHV 的前 4 个 8B 容器定义为机器学习的操作数,每个元素大小为 1B,最多支持 32 个参数,对于机器学习模型来说参数量较小,接下来的工作将扩展 PHV 使其支持更大的参数量,并在可编程数据平面中映射更复杂的神经网络模型(如卷积神经网络与循环神经网络),以支持更复杂的应用场景。

参 考 文 献

[1] Bosshart P, Gibb G, Kim H S, et al. Forwarding metamorphosis: Fast programmable match-action processing in hardware for SDN. ACM SIGCOMM Computer Communication Review, 2013, 43(4): 99-110

[2] Nick M, Tom A, Hari B. OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74

[3] Gebara N, Lerner A, Yang M, et al. Challenging the stateless quo of programmable switches//Proceedings of the 19th ACM Workshop on Hot Topics in Networks. New York, USA, 2020: 153-159

[4] Fei Jiawei, Ho Chenyu, Sahu A N, et al. Efficient sparse collective communication and its application to accelerate distributed deep learning//Proceedings of the 2021 ACM SIGCOMM Conference. New York, USA, 2021: 676-691

[5] Sapio A, Canini M, Ho C-Y, et al. Scaling distributed machine learning with in-network aggregation//Proceedings of the 18th USENIX Symposium on Networked Systems Design and Implementation. Virtual Event, USA, 2021: 785-808

[6] Yang Fan, Wang Zhan, Ma Xiaoxiao, et al. SwitchAgg: A further step towards in-network computing//Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. CA, USA, 2019: 185

[7] Lao C L, Le Y, Mahajan K, et al. ATP: In-network aggregation for multi-tenant learning//Proceedings of the 18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21). Virtual, USA, 2021: 741-761

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

[9] Xiong H, Zilberman N. Do switches dream of machine learning? Toward in-network classification//Proceedings of the 18th ACM Workshop on Hot Topics in Networks (HotNets'19). New York, USA, 25-33

[10] Zilberman N, Audzevich Y, Covington G A. NetFPGA SUME: Toward 100 Gbps as research commodity. IEEE Micro, 2014, 34(5): 32-41

[11] Anurag A, Kim C. Intel Tofino2—A 12.9 Tbps P4-programmable ethernet switch//Proceedings of the 2020 IEEE Hot Chips 32 Symposium (HCS). CA, USA, 2020: 1-32

[12] Yang M, Baban A, Kugel V, et al. Using Trio: Juniper networks' programmable chipset-for emerging in-network applications//Proceedings of the ACM SIGCOMM 2022 Conference (SIGCOMM'22). Association for Computing Machinery. New York, USA, 633-648

[13] Swamy T, Rucker A, Shahbaz M, et al. Taurus: A data plane architecture for per-packet ML//Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems. Lausanne, Switzerland, 2022: 1099-1114

[14] Akem A T-J, Bütün B, Gucciardo M, Fiore M. Henna: Hierarchical machine learning inference in programmable switches//Proceedings of the 1st International Workshop on Native Network Intelligence. Rome, Italy, 2022: 1-7

[15] Xie Guorui, Li Qing, Dong Yutao, et al. Mousika: Enable general in-network intelligence in programmable switches by knowledge distillation//Proceedings of the IEEE Conference on Computer Communications. London, UK, 2022: 1938-1947

[16] Zhou Guangmeng, et al. An efficient design of intelligent network data plane//Proceedings of the 32nd USENIX Security Symposium (USENIX Security 23). Anaheim, USA, 2023: 6203-6220

[17] Rashelbach A, Rottenstreich O, Silberstein M. A computational approach to packet classification//Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication. Virtual, USA, 2020: 542-556

[18] Tan J, Lv G F, Qiao G J. MBitTree: A fast and scalable packet classification for software switches//Proceedings of the IEEE Symposium on High-Performance Interconnects. Virtual Event, 2021: 60-67

[19] Li W, Yang T, Rottenstreich O, et al. Tuple space assisted packet classification with high performance on both search and update. IEEE Journal on Selected Areas in Communications, 2020, 38(7): 1555-1569

[20] Zhang X, Xie G, Wang X, et al. Fast online packet classification with convolutional neural network. IEEE/ACM Transactions on Networking, 2021, 29: 2765-2778

[21] Martín I, et al. Machine learning-based routing and wavelength assignment in software-defined optical networks. IEEE Transactions on Network and Service Management, 2019, 16(3): 871-883

[22] Sacco A, Esposito F, Marchetto G. RoPE: An architecture for adaptive data-driven routing prediction at the edge. IEEE Transactions on Network and Service Management, 2020, 17(2): 986-999

[23] Sun Weifeng, Wang Zun, Zhang Guanghao. A QoS-guaranteed intelligent routing mechanism in software-defined networks. Computer Networks, 2021, 185: 107709

[24] El-Garoui L, Pierre S, Chamberland S. A new SDN-based routing protocol for improving delay in smart city environments. Smart Cities, 2020, 3: 1004-1021

[25] Owusu A I, Nayak A. An intelligent traffic classification in SDN-IoT: A machine learning approach//Proceedings of the IEEE International Black Sea Conference on Communications and Networking. Odessa, Ukraine, 2020: 1-6

[26] Li H, Gao W, et al. Multiobjective bilevel programming model for multilayer perceptron neural networks. Information Sciences, 2023, 642: 119031

[27] Tan Lizhuang, Su Wei, Zhang Wei, et al. In-band network telemetry: A survey. Computer Networks, 2021, 186: 107763

[28] Gupta N, Jindal V, Bedi P. A survey on intrusion detection and prevention systems. SN Computer Science, 2023, 4(5): 439

[29] Emmerich P, Gallenmüller S, Raumer D, et al. MoonGen: A scriptable high-speed packet generator//Proceedings of the 2015 Internet Measurement Conference. Tokyo, Japan, 2015: 275-287

LIU Zhong-Pei, Ph. D. candidate. His current major research interest is computer network.

YANG Xiang-Rui, Ph. D., assistant professor. His current major research interest is programmable network.

YANG Ling, Ph. D. candidate. His current major research interest is computer architecture.

GAO Yuan-Hang, M. S. candidate. His current major research interest is computer network.

LÜ Gao-Feng, Ph. D., associate professor. His current major research interest is FPGA network accelerator.

WANG Bao-Sheng, Ph. D., professor. His current major research interest are network security.

SU Jin-Shu, Ph. D., professor. His current major research interests include computer network and network security.

Background

With the advent of the era of large AI models, large models with hundreds of billions or even trillions of parameters have emerged, and the computing tasks of some distributed applications can be offloaded to the NICs or switches of high-speed networks to potentially improve the performance of distributed applications and play a key role in the network. For

example, in distributed large model training and reasoning, offloading some aggregation computing functions in switches or NICs can effectively reduce a large amount of communication overhead generated during distributed computing, so that data can be calculated or aggregated synchronously in the process of network transmission, without introducing significant delay and throughput overhead. The current P4 oriented programmable network not only makes the network protocol more software-defined, but also enables the network to provide services for distributed application acceleration with certain computing power. However, the current typical programmable data plane architectures such as protocol independent PISA are not efficient enough in matrix computation. The key reason for our analysis of this defect is that the multi-instruction multi-data stream mode based on very long instruction words in PISA architecture is not suitable for the situation where a large number of calculations are required and all computing units perform the same operation, which is costly and slow compared to the single-instruction multi-data stream mode. Focusing on this problem, we propose CAInNet, an AI-accelerated in-network computing model. On the basis of supporting all P4 programmable data plane capabilities, the platform integrates SIMD and MIMD two computing modes, so that the switch and NIC can not only support protocol-independent network packet processing, but also accelerate the packet data carrying AI inference and training in the data transmission process. In order to demonstrate the capability and effect of CAInNet in in-network computing and network programmability, we use in-band network telemetry to realize network visualization, and deploy a multi-layer perceptron model to realize AI-based packets classification instead of the traditional TCAM table lookup routing method. Experiments show that the accuracy of the packets classification method based on machine learning inference can reach 98.3% and save 98.7% of the memory space for 5 K routing entries, which can solve the routing explosion problem well.

This project was supported by the National Natural Science Foundation of China (62372462, subject name "Research on Framework and Core Mechanism of Domain-specific Network DSN"), the Natural Science Foundation of Hunan Province (2023JJ40682), the Youth Independent Innovation Science Foundation of NUDT (ZK2023-13), and the High-level Talent Funding Project of NUDT.