**ETCpipe: Efficient Encrypted Traffic Classification System Using DPU-Based Networking and GPU-Enhanced Preprocessing**

**Lingxiang Hu**

**Abstract**

随着网络流量模式的日益复杂化和加密通信技术的广泛应用，基于深度学习的实时流量分析需求急剧增长。现有深度学习网络模型虽在加密流量分类（ETC）中表现卓越，但其实现需消耗大量服务器资源。目前的解决方案，例如FENXI系统，尽管为ETC的实时部署提供了支持，但配置复杂并且侧重于特定模型，忽视了对ETC领域多数模型的支持，从而限制了其通用性。

为应对上述挑战，本研究提出了“ETCpipe”，一种专为ETC任务设计的执行系统。ETCpipe通过将流量管理和特征提取任务解耦至领域特定加速器，有效消除了配置复杂性，提升了处理性能，并释放了主机CPU资源。系统中的FlowManager组件将流量管理任务卸载至DPU，而特征提取任务则由GPU上的GPUFE组件承担。ETCpipe系统利用FlowManager作为网络流入点，将待分析数据交由GPUFE进行特征提取，最终通过推理引擎完成数据分析。我们还解决了GPU处理过滤后数据包的特征提取的难题，通过GPUdirect技术和设计并行算法，将数据直接传输至显存中，加速了特征提取过程。

在NVIDIA V100 GPU和BlueField-3 DPU平台上，我们实现了ETCpipe系统的原型，并验证了其高效性。系统性能评估从网络吞吐量和特征提取效率两个维度进行：与DPU自带的OVS相比，在多种任务上ETCpipe维持了高达100Gbps的网络吞吐率；与FENXI系统相比，在多种任务上ETCpipe的处理效率至少相当于三个CPU核心，并在FENXI的用例上远超3个CPU核心。

本研究的主要贡献在于为不同ETC任务提供了一个深度解耦的系统设计，并通过对原始数据包进行深入的特征提取与测试，不仅简化了系统配置，还显著提升了处理速度和系统的通用性，为ETC任务的实时部署提供了新的解决方案。

**Introduction**

在现代网络架构中，实时流量分析成为了关键组成部分，尤其是随着网络流量模式的复杂化和加密通信技术的广泛应用，对基于深度学习（DL）的流量分析方法的需求持续增加。特别是在加密流量分类（ETC）领域，新兴的深度学习网络模型已证明其卓越性能，成为处理关键网络流量分析任务的强大工具。然而，这些任务在传统服务器上执行时对资源的消耗极大，涉及流量管理、特征提取（亦称数据预处理）和模型推理等多个环节。随着网络速度的显著提升——超过100 Gbps——实时推理系统不仅需保持网络通信的流畅，更必须能够及时处理和分析庞大的流量数据。因此，如何在高速网络环境下有效部署ETC系统，成为了一个亟待解决的问题。

为应对实时分析的需求，业界已开发出多种解决策略。一种常见的方法是将处理任务卸载至网络设备，如n3ic项目通过在SmartNIC上部署模型直接在数据平面执行流量分析，从而实现在网计算。这种方法的优势在于显著减少跨设备通信并释放CPU资源，但由于SmartNIC的资源限制，可部署的模型受到较大约束。另一类解决方案则侧重于使用专用推理硬件，如FENXI系统，它代表了当前技术的最前沿（State-of-the-Art, SOTA）。FENXI通过解耦网络数据平面和流量分析任务，并构建高效的处理流水线，显著提高了系统性能。此方案利用多核CPU执行流量管理和特征提取，同时使用GPU或TPU进行深度学习模型的推理。尽管FENXI能支持较为复杂的模型，其配置的复杂性却成为了一个显著的问题：流水线的资源配置需要手动调整，且要求系统管理员对硬件性能及模型需求有深入的了解以实现最佳系统性能。此外，尽管FENXI进行了基于数据包大小的测试，它在处理原始数据包负载这种更为重型的任务时的通用性尚未得到验证。

在本研究中，我们开发了“ETCpipe”，一个专门为处理异构加密流量分类（ETC）任务设计的执行系统。该系统的核心创新在于其将流量管理和特征提取任务进一步解耦到领域特定加速器上，这一策略显著释放了主机CPU的资源，同时解决了系统配置的复杂性，极大提高了处理速度。更具体地说，我们将流量管理任务卸载到了SmartNIC上的FlowManager。FlowManager负责筛选和过滤数据包，将需要进一步分析的数据包传递给GPU。在GPU上，名为GPUFE的特征提取组件负责提取流量数据的特征信息，最终通过推理引擎（例如针对NVIDIA显卡优化的TensorRT）完成深度学习分析任务。

这种解耦虽然提供了显著的优势，但也带来了一些技术挑战。尽管GPU的高度并行处理能力使其在收集到足够数量的数据包后能有效进行特征提取和推理，它在处理过滤后的数据包时却面临困难，因为GPU本身不擅长直接执行流重组这一任务。这一局限性要求CPU介入以完成数据流的重组工作，随后再将数据拷贝至显存中，这不仅未能完全卸载流量管理任务，还增加了额外的内存拷贝过程，影响了系统效率。

为了克服这些技术挑战，我们优化了数据传输方式并设计了专用的并行处理算法。首先，我们采用了GPUdirect技术，直接将过滤后的数据包从网卡传输至显存中，绕过CPU主存，以减少数据处理的延迟。同时，我们应用了排序和前缀扫描等并行算法完成数据流的重组，实现了高效的特征提取。在NVIDIA V100 GPU和BlueField-3 DPU平台上，我们开发了ETCpipe系统的原型，以网络吞吐量和特征提取效率为关键性能指标进行评估。

通过运用iperf工具进行端到端测试，我们验证了ETCpipe系统在执行多种任务时对网络吞吐量无负面影响，这得益于其高度并行的系统设计。进一步的，我们利用pktgen-dpdk生成的数据包对系统的特征提取效率进行了测评。测试结果显示，在执行多种ETC任务时，ETCpipe的特征提取效率至少达到了FENXI CPU解决方案的三至四倍。针对FENXI的研究用例，ETCpipe的效率远超过三个CPU核心的性能，从而显著提升了整体处理能力。

本研究的主要贡献可以概括为以下两点：

（1）系统设计的创新：本研究首次实现了将CPU承载的任务深度解耦至领域特定加速器的系统设计，显著简化了当前系统中存在的配置复杂性。这一设计不仅优化了资源利用效率，还提高了整个系统的处理速度。

（2）原始数据包负载模型的分析：我们提出的框架是首个针对基于原始数据包负载进行深入分析的系统，有效地填补了现有研究中对此类分析缺乏的不足。通过对原始负载数据的直接分析，我们的框架能够提供更精准的流量分类和安全分析，从而增强了网络监控和管理的能力。

总之，我们提出了ETCpipe，一个加密流量分类系统解决方案，旨在应对高速网络环境中实时数据处理的挑战。ETCpipe通过领域特定加速器对流量管理和特征提取任务进行了深度解耦，有效释放了CPU资源，同时简化了系统的配置流程。此外，ETCpipe采用了先进的技术如GPUdirect来优化数据传输，提高了数据处理的速度和效率。通过在V100 GPU和BlueField-3 DPU平台上的实验验证，我们证明了ETCpipe能够在不影响网络吞吐量的情况下，显著提高特征提取的效率。这样的系统设计不仅提升了性能，还增强了对多种ETC任务的支持能力，为部署各类ETC深度学习模型提供了一个通用且灵活的解决方案。

**Background and Motivation**

2.1 深度学习在ETC的应用

深度学习加密流量分类（ETC）领域根据输入特征不同，可以分为基于数据包的原始信息和原始有效载荷两种。前者能够基于数据包的原始信息（如数据包大小、到达时间以及数据包内容），例如，研究工作DF采用卷积神经网络（CNN）从加密流量的数据包大小序列作为输入；FS-Net利用循环神经网络（RNN）进行相似的任务；后者直接以原始有效载荷作为输入特征，例如Deeppacket和TSCRNN，直接以数据包的原始有效载荷作为输入。另外，由于大语言模型的蓬勃发展，大预言模型也被应用在了ETC任务中，最近的研究，如LLM和ET-BERT，也在ETC任务中证明了深度学习模型的出色性能。

2.2 DPU 和 GPU

数据处理单元（DPU）是专为网络数据处理而设计的高度专业化硬件，其中集成了如ARM处理核心的专用加速器。DPU利用GPUdirect技术加速内存拷贝，PUdirect技术允许数据从网络接口卡（NIC）直接传输到GPU内存，绕过CPU，从而提高数据处理速度。此外，DPU配备了接收端缩放（RSS）卸载功能，R接收端缩放（RSS）是一种分散网络流量的技术，它通过在网卡上计算数据包的五元组哈希来分配流量至多个CPU核心，从而提高处理效率。结构如图X所示

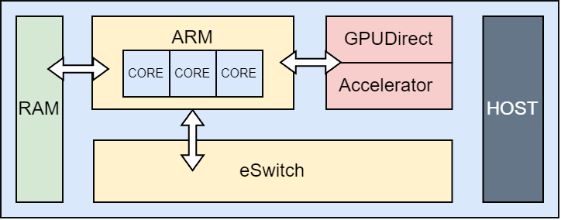


图 Bluefield-3 DPU Architectures

GPU的stream功能支持在同一硬件上并行执行多个任务，这使得不同的计算和数据传输操作能够同时进行，避免资源独占。通过这种方式，特征提取和模型推理等不同过程可以有效地并行部署在单个GPU上，显著提高了加密流量分类（ETC）任务的处理效率和响应速度。

2.3 Motivation

在我们的研究领域中，FENXI系统代表了实时加密流量分类（ETC）处理的一项重要进展。该系统通过CPU来执行流量管理和特征提取任务，并利用TPU/GPU加速模型推理，有效地解耦了数据包转发与模型分析任务。如下图所示，FENXI通常采用1:1:1的配置比例，在特定的研究案例中表现出色。

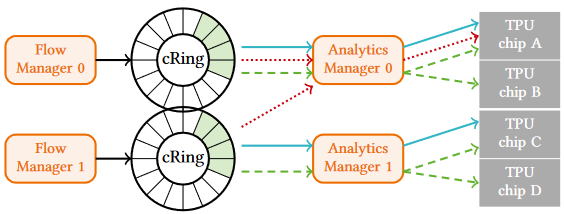


图 FENXI 配置流水线

然而，我们对多种深度学习ETC模型的本地推理实验揭示了系统的显著局限。不同模型因其结构和参数规模差异而导致运行时间的显著不同；此外，在PyTorch框架下，CPU与GPU的推理速度差异也对模型的执行时间产生了显著影响。这要求系统管理员不仅要深入了解各模型的特点，还需熟悉推理硬件和软件环境，以免不恰当的配置成为性能瓶颈。例如，对于推理时间较短的Darknet模型，可能需要更多的CPU核心来处理特征提取；而对于推理时间较长的TSCRNN模型，配置过多的CPU核心则可能造成资源浪费。如图所示

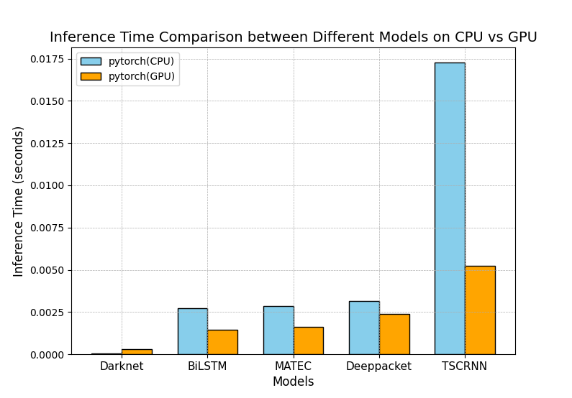


图 不同模型在不同平台的执行推理

此外，FENXI的研究主要限于分析前K个数据包的长度，没有考虑基于原始负载数据的更多样化分析。这种研究的局限性削弱了模型处理原始负载数据时的适用性，并限制了系统的广泛应用。原始负载数据是ETC任务中的关键输入特征，其深入分析对理解加密流量的本质至关重要。

因此，FENXI系统配置的复杂性和在通用性上的不足，是我们研究的动机。

**Systems Design**

3.1 设计目标

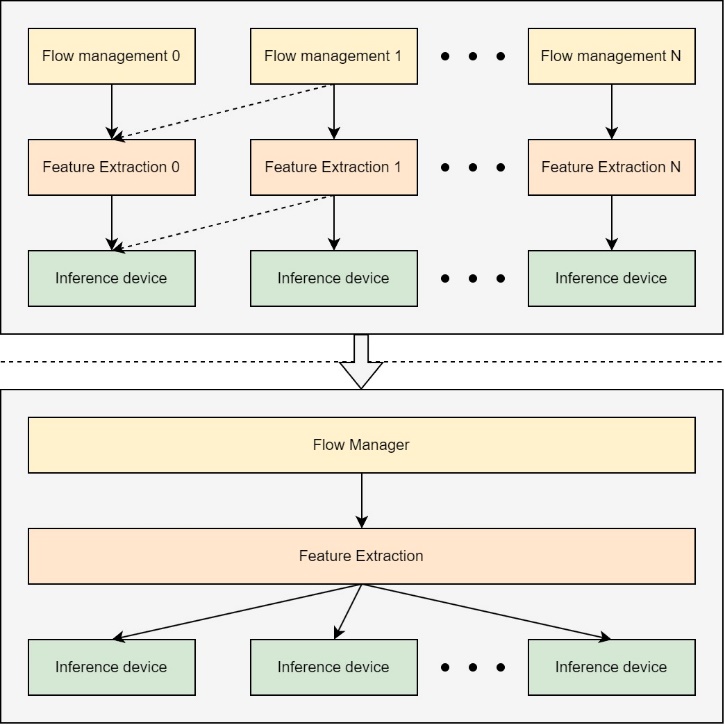


图 主要观点

本研究旨在彻底解耦流量管理与特征提取任务，并最终通过集成硬件加速器并利用推理框架，实现一个由深度学习驱动的加密流量分类（ETC）系统。此解耦策略的提出基于对现有ETC系统配置复杂性的关键分析，特别是在系统需要适应多样化深度学习模型的环境中。

从动机中我们可以得出，传统ETC系统因配置的难题而难以灵活应对多变的模型需求。例如，当推理模型或特征提取算法需要更新时，整个系统的配置过程需要重新进行，这不仅耗时且易于出错，还大幅增加了系统的维护负担。为了解决这一问题，本研究提出将流量管理任务专门交由DPU处理，并将特征提取及模型推理任务专门交由GPU执行。这种硬件层面的解耦不仅简化了系统配置，提高了处理效率，也增加了系统对不同模型调整的灵活性。

此外，该解耦方法还为系统的扩展性和可升级性提供了便利。各硬件组件可以独立进行优化和升级，无需担心对整个系统造成干扰。因此，这种策略直接解决了现有系统中存在的通用性和配置复杂性问题。

首先，我们设计需要实现快速特征提取，这需要在特征提取过程中实现计算优化和数据传输的效率。在算法层面，我们计划开发优化的计算方法，采用并行处理策略以加速数据处理流程。在数据传输方面，改进传输的方式，降低延迟和带宽占用，实现更高效的特征构建。

同时，为了提升系统对各类ETC应用的适用性，我们将扩展特征分析至基于原始数据包负载，填补现有系统在原始数据处理方面的空白，并确保即使在系统升级活ETC任务改变后依然无需关注配置问题，减少管理员的干预需求。

在数据中心部署系统时，我们的设计需要确保系统能够处理高吞吐量的数据流，从而维护网络服务质量（QoS）和优化用户体验。为此，我们专门在数据处理单元（DPU）上开发高效的算法和组件，这些设计将网络数据管理与推理任务有效分离。通过这种方式，我们确保系统的数据处理和模型推理操作不会相互干扰，从而保持不变的高网络传输速率和响应能力。

3.2 系统架构概述

A diagram of a computer

Description automatically generated

图 ETCpipe 系统架构

本节介绍ETCpipe系统的架构，该架构专门设计来解耦流量管理和特征提取任务，提高处理效率保证网络正常。系统主要由三个部分构成：流量管理器（FlowManager）、特征提取器（GPUFE）、以及推理引擎。FlowManager负责在数据处理单元（DPU）上筛选和管理数据流；GPUFE在图形处理单元（GPU）上执行特征提取；推理引擎用于进行分析得到结果。

首先，网络数据包经过流量管理器，我们将该组件命名为FLowManager，该组件部署在DPU，负责接收并按照特定条件过滤数据包。根据数据包的五元组信息，ETCpipe能够识别并追踪流，如果是符合需求的数据包，FLowManager将给特征提取器进行转发，该流程有效解耦了数据分析任务与数据包转发过程，确保了系统的高吞吐。

FLowManager转发的符合条件的数据包将进入GPUFE，一个部署在GPU上的特征提取器。该组件利用GPUDirect技术，直接从流管理器将数据包高效传输至GPU显存，从而降低延迟并提高数据处理速度。GPUFE并行执行特征提取任务，并利用专门的算法从数据包中提取所需特征。更多关于特征提取算法的细节将在后续章节进行详细讨论。

提取的特征随后被送入推理引擎进行分析，以生成最终结果。在本系统中，推理引擎可部署于CPU、GPU或TPU上，但考虑到我们的原型是基于Nvidia V100 GPU，我们选择使用TensorRT推理框架，它优化了基于Nvidia GPU的推理性能。

综上所述，ETCpipe系统处理数据包的流程概述如下：首先，数据包首先经过至FlowManager，FlowManage判断数据包是否符合预设的条件；那些不符合条件的数据包会被直接转发，而符合条件的数据包则选定进行进一步的分析。这些数据包随后被传递至GPUFE，之后GPUFE进行提取特征。提取的特征随后输送至推理引擎，负责进一步的数据分析与结果生成。这种系统的解耦设计，极大地简化了管理层面的操作复杂性，允许管理人员专注于数据分析，增强了系统的灵活性和效率。

**4. Implementation**

**4.1 FlowManager**

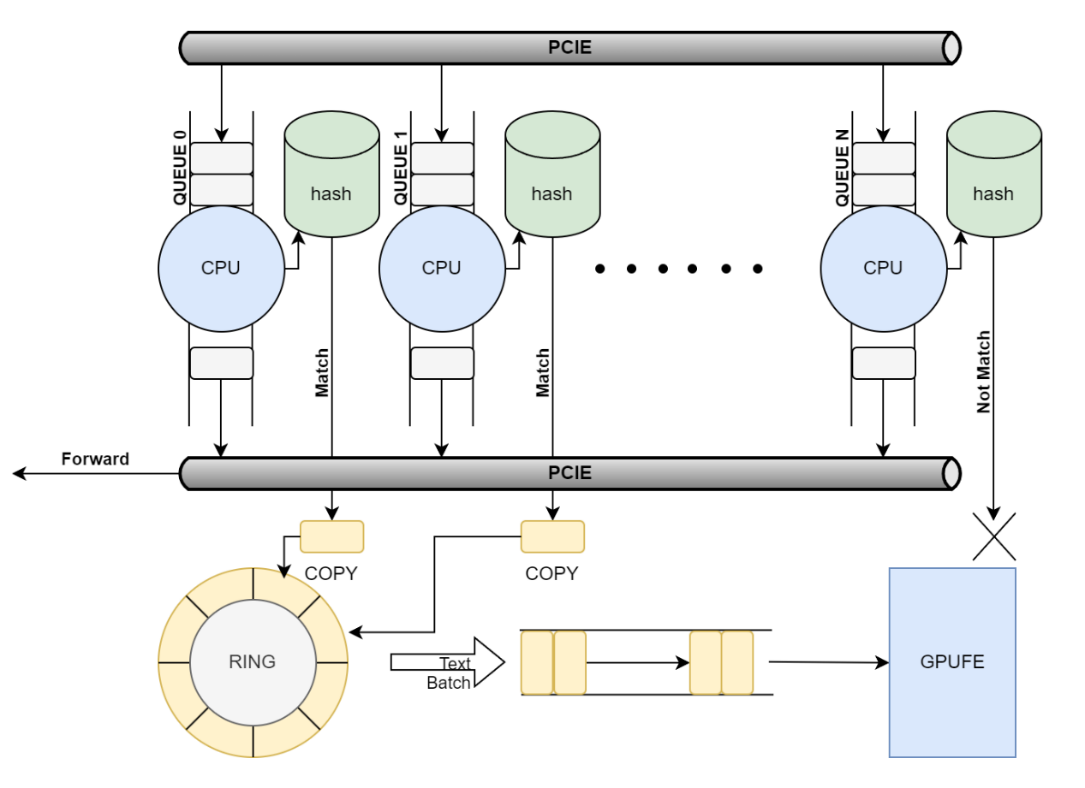


图 FlowManager 框架

FlowManager组件是ETCpipe系统处理的起点，负责确保网络通信的无缝和高效率。在执行流量管理的同时，它对进入系统的流量进行状态跟踪和必要的过滤，为GPUFE组件的特征提取任务做前置准备。图X展示了流量到达FlowManager的初始处理步骤：数据包首先被均匀分配至多个处理队列，然后根据预定义的过滤规则进行筛选。符合条件的数据包被复制到DPU上的RING队列，并以批量方式传输到GPUFE。本节将详细阐述FlowManager的设计和实现细节。

FlowManager的核心任务是实时流量状态跟踪。为适应各种ETC任务的输入需求多样性，例如，某些任务可能只需要流的前N个非ACK-FIN数据包，FlowManager利用DPU提供的广泛计算资源（包括通用计算资源和专用硬件加速器）进行高效的处理。它通过部署多个ARM核心，构建了一个高效的多处理队列体系。每个ARM核心分别管理一个入队列和出队列，并维护一个以五元组为键、流状态为值的哈希桶。网络接口卡（NIC）的接收端口规模集（RSS）功能根据数据包的五元组信息计算哈希值，并将数据包均匀分散到不同处理队列。这确保数据包分布的均衡性和处理的并行性，显著减少了单个处理队列的负载并提升了整体吞吐能力。

最终，这一策略使FlowManager能够支撑高速的流量分析，同时保持网络通信的低延迟和高可靠性。FlowManager的性能和效率将在第7章的端到端测试中进行详细评估，以验证其在实际网络环境中的表现和对整个ETCpipe系统性能的贡献。

**6.** **GPUFE**

在ETCpipe系统中，特征提取环节是将原始网络流量转换为深度学习模型可解析的输入特征的关键步骤。继FlowManager完成对数据流的初步筛选与识别后，GPUFE模块负责进一步对这些数据进行预处理，并转换为模型可用的输入特征，从而为推理过程提供必要的数据输入。为了充分展示ETCpipe系统如何填补了依赖原始数据包负载进行特征提取的研究空白，本章节将专注于基于原始数据包负载的特征提取过程，探讨传统基于CPU的特征提取实现的局限性、采用GPU加速所面临的挑战，以及GPUFE模块如何有效解决这些问题。

6.1基于CPU的特征提取

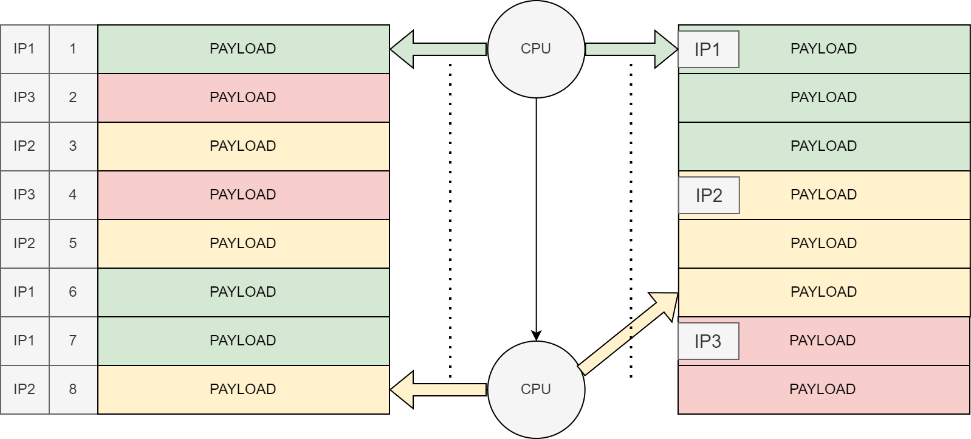


图 基于CPU的特征提取（着重展示基于原始数据负载）

在之前的工作中，特征提取通常通过CPU完成。如图展示了一个典型的基于CPU的特征提取过程，其中以8个数据包为例。在这一过程中，CPU需逐一处理接收到的数据包。由于数据包是按不同流穿插到达的，CPU必须从第一个数据包开始，串行地识别各个流并提取特征。一旦某个流的特征提取完成，该流的特征便被送往后续的推理模块。尽管部署多个CPU核心能在一定程度上提升处理性能，但如前所述，设计有效的流水线配置对于系统管理员而言是一大挑战。这不仅要求管理员对深度学习模型的需求和硬件特性有深入的理解，还需要他们综合考虑以最大化资源利用率。

此外，基于CPU的特征提取方法在处理高速网络流量时面临着显著的性能瓶颈。数据包的串行处理机制限制了处理速度，尤其是在数据流量激增的情况下，需要分配更多的CPU资源用于特征提取，这进一步增加了系统配置的复杂性。

综上所述，尽管基于CPU的特征提取方法在早期的ETC系统中得到了广泛应用，但其在处理效率、资源利用和系统配置方面的限制，使得寻求更高效的特征提取方案成为ETC领域的迫切需求

6.2GPUFE

6.1 技术挑战及解决策略

在探索GPU加速特征提取过程中，我们面临了一系列技术挑战，尤其是关于流重组和资源管理的问题。以下是详细分析及针对性解决方案。

**流重组的挑战**：特征提取任务的核心是流的识别与重组，如图所示，这一过程要求精确地根据流的顺序进行数据重组和特征提取。尽管基于CPU的处理策略可以通过串行处理实现流重组，但此方法难以直接迁移到GPU上，因为直接采用与CPU相同的算法会引入两个主要问题。首先，为了保证数据包的顺序处理，需要在GPU上实施锁机制，这将大幅降低GPU的并行处理能力。其次，从CPU向GPU拷贝数据包也会显著影响性能，由于数据传输的开销。

**资源抢占的挑战**：特征提取是系统中的关键组件，但是推理引擎可能使用GPU资源导致竞争。从特征提取的性能角度考虑，虽然使用持久化kernel和更多的线程块会提升特征提取效率，但这样做可能会导致推理引擎在访问GPU资源时遇到竞争，影响系统的整体性能。因此，设计时必须在保证足够性能的同时，合理分配GPU资源，避免过度占用。

**解决策略：**为应对上述挑战，我们采取了以下措施：

资源限制：通过对GPU资源的精细管理，当检测到有1024个数据包到达时，我们启动一个包含1024线程的线程块来处理这些数据包。这种方式既充分利用了GPU的并行处理能力，又避免了资源的过度占用。

GPU Direct技术：利用GPU Direct技术，实现了数据包从网络接口卡（NIC）直接传输到GPU内存的过程，减少了CPU到GPU的数据拷贝需求。这一策略显著降低了数据传输的开销，为高效的并行算法实施提供了基础。

并行算法设计：在GPU上，我们开发了专门的并行算法来实现流重组。这一算法针对GPU的计算模型优化，最大化了流重组和特征提取过程的效率。我们在6.2详细的讨论算法的设计与实现。

6.2GPUFE组件实现

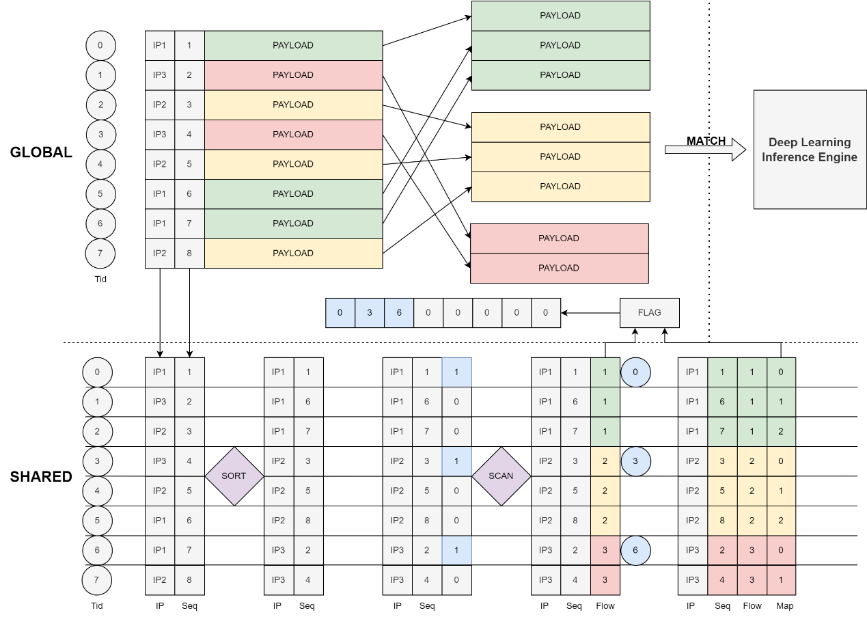


图 基于GPU的特征提取（着重展示基于原始数据负载）

当待处理的数据在GPUFE上准备完成后，我们映射global内存中数据包的IP头部和位置ID（通过线程号+1得到）复制到shared内存上，这避免了后期数据包负载的频繁拷贝。

之后在共享内存中我们对IP头部进行处理。如图所示我们计算哈希与偏移计算：每个数据包对应的处理线程计算出基于流ID的哈希值，指向GPU显存中为该数据流特征预留的存储区（ptr）。结合MAP记录的序号，进一步计算出在存储区内的偏移量（offset），确立数据包特征的具体存储位置。特征向量存储：完成上述计算后，特征向量按计算得到的地址（ptr + offset）存储，确保每个数据流的特征向量都能在GPU显存中有序地并行生成。特征向量并行生成后，下一步是将这些数据送入推理引擎。为此，我们设计了一个高效的数据流转移策略，确保数据流能够无缝地从特征提取转向推理阶段。特征批次传递：一旦一组数据流的所有数据包特征提取完成，其存储地址立即被传回CPU。CPU随后负责管理这些特征数据，将其送入推理引擎。

选择TensorRT推理引擎：鉴于其针对GPU优化的高性能特性，我们选用TensorRT作为推理引擎。TensorRT支持广泛的模型和框架，适应高吞吐量需求，有效减少了数据传输开销，保障了推理过程的高效性和低延迟。

通过上述并行化特征映射与推理准备过程，ETCpipe系统能够有效地处理高速传入的网络流量数据，将其转换为深度学习模型可用的输入特征，并为快速准确的推理决策做好准备。

**7.Evaluation**

本研究的验证分为两个主要部分，旨在评估FlowManager和GPUFE两个核心组件的性能。我们通过实验验证ETCpipe系统部署ETC任务时对网络通信的影响，以及GPUFE组件的处理效率。

**实验设置**：实验环境基于Dell服务器，配置包括NVIDIA V100 GPU和BlueField-3 DPU，操作系统为Ubuntu。我们的实验代码已开源于GitHub: https://github.com/CHRIS123540/GPU。

**端到端测试**：端到端的性能测试采用Iperf工具，旨在识别系统处理瓶颈，并评估网络设备队列数据包积压对网络协议的影响。相较于先前工作中使用Pktgen的方法，Iperf能够提供更为直接的网络带宽测试结果，从而准确评估FlowManager组件是否对网络通信产生影响。

**GPUFE组件测试**：在GPUFE的性能测试中，我们选用Pktgen-dpdk作为发包工具，以消除网络协议可能引入的性能干扰，如乱序或重传，从而更专注于评估特征提取的效率。此部分测试关注于基于原始数据包负载的特征提取——一个之前研究中未充分探讨的领域。我们还将复现基于CPU的特征提取方案，并与GPUFE组件的性能进行对比，展示后者在处理能力上的优势。

7.2端到端测试

端到端性能测试的结果如图所示，我们关注的是在应用特定过滤规则（选定流的前N个数据包，其中N=10）下，FlowManager与DPU上原有的Open vSwitch（OVS）的性能对比。测试旨在评估ETCpipe系统对现有网络通信的潜在影响。通过在Iperf测试中模拟2至128个并发流，结果显示FlowManager利用ARM多核处理能力及硬件加速特性，实现了比原始OVS更高的吞吐率。当并发流量超过128时，系统对网络通信的影响变得不显著，表明ETCpipe系统在高负载情况下能够维持网络通信的效率。

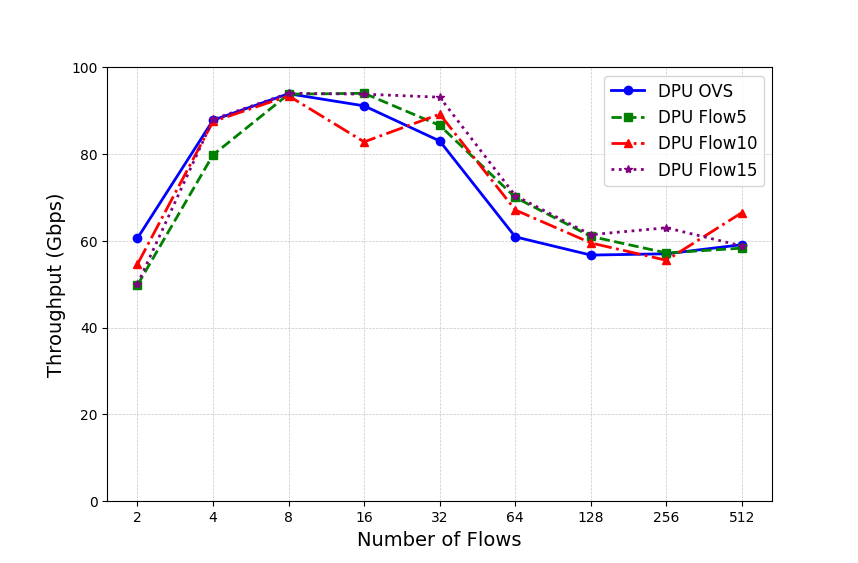


图 e2e测试

7.3 GPUFE 的处理性能

如图所展示的GPUFE与CPU处理性能的对比测试结果，我们选取的特征是[Height,Len]，其中Height是数据包数量取10，Len是需要提取的数据包长度取1000，这一设置基于当前研究中常见的数据包负载场景。通过Pktgen-dpdk工具，我们生成了从10条到100条流的特征提取任务，并测量了使用CPU与GPUFE两种方式的执行时间。结果表明，GPUFE的处理能力至少相当于3个CPU核心。性能的提升归因于GPU的多核并行加速能力和内存拷贝次数的减少，这一发现突出了ETCpipe在特征提取任务上的高效率。

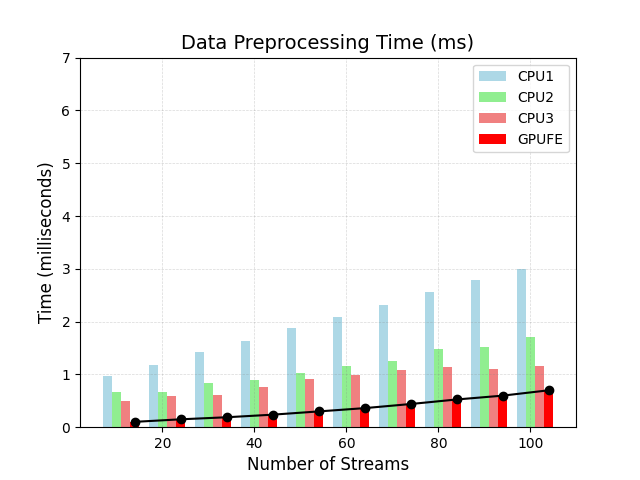
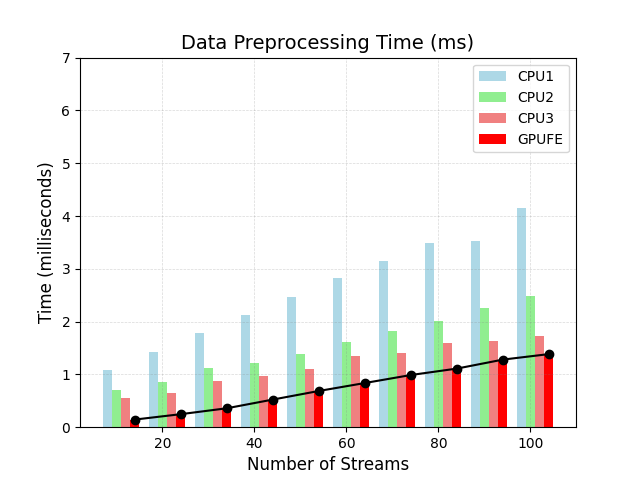
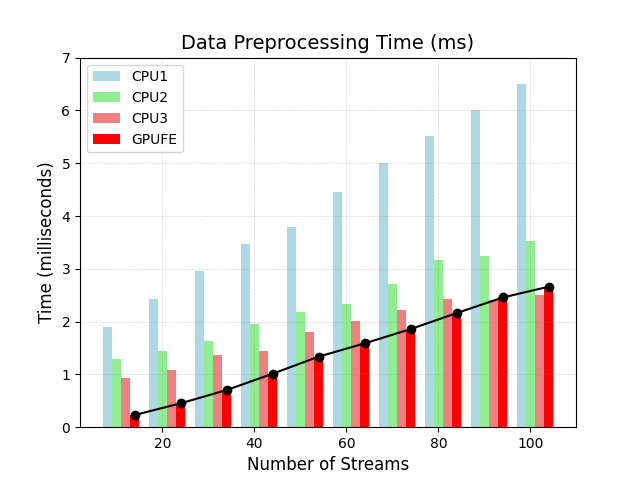


图 GPUFE VS CPU

**第七章 Future Work**

在本研究中，我们展示了DPU的潜在能力，特别是在实现高性能网络通信方面。现有的FlowManager通过结合ARM处理能力和RSS硬件卸载技术，有效地支持了系统的网络处理需求。然而，NVIDIA BlueField-3 DPU提供了专门的硬件接口（称为DOCA硬件语言），它支持基于eSwitch的Match-Action硬件处理管道，这一方案相比于使用ARM处理相同任务能够带来更高的效率。已有的研究表明，通过利用DOCA管道，可以在Linux TC（Traffic Control）上实现100Gbps的流量管理。未来，我们计划深入探究DOCA技术，以期提升FlowManager的处理能力和效率。

此外，ETCpipe系统目前采用TensorRT引擎作为推理后端，专门针对NVIDIA GPU进行优化。虽然TensorRT提供了出色的推理性能，但考虑到系统设计的多样性和扩展性，我们预计将探索接入其他推理引擎。虽然这可能不会为特定的硬件配置带来最佳性能，但这一步骤对于使系统能够适应更广泛的应用场景和推理平台具有重要意义。通过增加对多种推理引擎的支持，ETCpipe将能够更加灵活地应用于不同的环境中，为系统的广泛部署和应用奠定基础。。

**第八章 Conclude**

本研究成功开发了ETCpipe，一个针对加密流量分类（ETC）任务设计的创新系统，通过利用领域特定加速器深度解耦流量管理与特征提取任务，显著提高了处理效率。ETCpipe采用SmartNIC（FlowManager）和GPU（GPUFE）分别处理这些任务，有效释放了主机CPU资源，简化了系统配置，同时实现了加速数据处理。ETCpipe的设计优化了流量过滤、特征提取和推理执行的每个步骤。FlowManager利用SmartNIC技术实现快速流量管理，而GPUFE则利用GPU并行处理能力高效提取特征。此外，通过GPUdirect技术和高效算法减少了CPU和GPU间的数据拷贝，进一步优化性能。

测试结果显示，ETCpipe在保持网络吞吐量的同时，能提供比现有CPU解决方案高达三至四倍的特征提取速度，证明了其在效率和性能上的显著提升。ETCpipe不仅技术上取得进步，还在原始数据包负载分析等实际应用中展示出潜力，为网络监控和管理提供更精准的流量分类和安全分析能力。

总而言之，ETCpipe为实时流量分析领域带来了创新，填补了现有技术的空白，为未来研究和开发奠定了基础，预示着对该领域的深远影响。

**Ref**

DPU链接

https://resources.nvidia.com/en-us-accelerated-networking-resource-library/datasheet-nvidia-bluefield