



专用数据处理器（DPU） 性能基准评测方法与实现

DPU Benchmark Methodology and Implementation

技术白皮书

第 1.0 版

2022 版 7 月

主编单位

中国科学院计算技术研究所

中科驭数（北京）科技有限公司

联合编写发布单位

处理器芯片全国重点实验室

中国计算机学会（CCF）集成电路设计专业委员会

中国计量测试学会集成电路测试专业委员会

编委会

专家顾问

- 孙凝晖、李晓维、陈云霄、詹剑锋

编委成员

- 主编：郝贵海
 - 编委（按姓氏拼音排序）：陈岩、程丽云、樊海爽、耿嘉、顾帅、黄明亮、林涵越、卢文岩、廖云坤、李明、罗梅、刘昱泽、孙伟、吴婧雅、袁晓飞、赵汝杰、赵巍岳、赵越、张社广、张宇、张宇军
-

致谢

本白皮书编写过程中还得到了中国电子技术标准化研究院、中国信通院技术和标准研究所、北京市经信局，国际测试委员会（International Open Benchmark Council），曙光信息产业有限公司、浪潮集团、英伟达、天翼云、中国移动研究院基础所、中国移动研究院网络与 IT 技术研究所、上交所技术有限公司、深圳交易所、中国电信股份有限公司广州研究院、中移（苏州）软件技术有限公司、中兴通讯、京东云计算有限公司、百度、腾讯 TEG 事业群数据平台部、华泰证券、Ucloud、世纪互联、H3C、恒生电子、人大金仓、DaoCloud, SmartBI 等机构和上下游企业的支持与协助，特此表达衷心的感谢！

版权声明

本白皮书版权属于主编和联合编写发布单位，并受法律保护。转载、摘编或利用其它方式使用本白皮书文字或者观点应注明“来源: 专用数据处理器 (DPU) 性能基准评测方法与实现, 中国科学院计算技术研究所/中科驭数, 鄢贵海等”。违反上述声明者，版权方将追究其相关法律责任。

目录

第 1 章 DPU 性能评测导论	1
1.1 DPU 评测的挑战和重点	1
1.2 开展 DPU 评测建议标准	3
1.3 系统测试的四个维度	4
1.3.1 网络维度	4
1.3.2 存储维度	5
1.3.3 计算维度	6
1.3.4 安全维度	7
1.4 内容的组织结构	8
第 2 章 DPU 性能评测系统框架与测试流程	9
2.1 DPU 测试系统 (SUT)	9
2.1.1 单端型 (Single-End) 测试系统	10
2.1.2 端到端型 (End-to-End) 测试系统	11
2.1.3 多端型 (Multi-End) 测试系统	13
2.2 测试要求	14
2.3 测试活动	15
2.3.1 测试选择	15
2.3.2 测试准备	15
2.3.3 测试执行	15
2.3.4 测试报告	16
第 3 章 面向网络的基准评测	17
3.1 基础网络加速	17
3.1.1 基于 DPU 的基础网络加速结构	18
3.1.2 测试系统	18
3.1.3 Benchmark	20
3.2 RDMA	22
3.2.1 基于 DPU 的 RDMA 加速结构	22

3.2.2	测试系统	23
3.2.3	Benchmark	24
3.3	虚拟交换机/路由器卸载加速	26
3.3.1	基于 DPU 的 OVS 加速结构	26
3.3.2	测试系统	27
3.3.3	Benchmark	27
3.4	SD-WAN 网关卸载加速	29
3.4.1	基于 DPU 的 SD-WAN 加速结构	30
3.4.2	测试系统	31
3.4.3	Benchmark	32
3.5	负载均衡 LB 卸载加速	33
3.5.1	基于 DPU 的 LB 加速结构	33
3.5.2	测试系统	33
3.5.3	Benchmark	34
3.6	5G MEC	36
3.6.1	基于 DPU 的 5G MEC 加速结构	37
3.6.2	测试系统	38
3.6.3	Benchmark	39
第 4 章	面向存储的基准评测	41
4.1	NVMe-oF 加速	41
4.1.1	基于 DPU 的加速结构	42
4.1.2	测试系统	44
4.1.3	Benchmark	45
第 5 章	面向计算的基准评测	49
5.1	Spark 加速	49
5.1.1	基于 DPU 的 Spark 加速结构	50
5.1.2	测试系统	50
5.1.3	Benchmark	51
5.2	AI 加速	52
5.2.1	基于 DPU 的 AI 加速结构	53

5.2.2	测试系统	54
5.2.3	Benchmark	54
5.3	数据库加速	57
5.3.1	基于 DPU 的数据库加速结构	57
5.3.2	测试系统	58
5.3.3	Benchmark	58
5.4	视频/图像编解码	60
5.4.1	基于 DPU 的视频/图像编解码加速结构	60
5.4.2	测试系统	61
5.4.3	Benchmark	62
5.5	云原生加速	65
5.5.1	基于 DPU 的云原生加速结构	65
5.5.2	测试系统	66
5.5.3	Benchmark	68
第 6 章	面向安全的基准评测	72
6.1	DPI	72
6.1.1	基于 DPU 的 DPI 加速架构	72
6.1.2	测试系统	73
6.1.3	Benchmark	74
6.2	防火墙	75
6.2.1	基于 DPU 的防火墙加速架构	75
6.2.2	测试系统	76
6.2.3	Benchmark	76
6.3	IPS/IDS	78
6.3.1	基于 DPU 的 IPS/IDS 加速架构	78
6.3.2	测试系统	78
6.3.3	Benchmark	80
6.4	IPSec	81
6.4.1	基于 DPU 的 IPSec 加速结构	81
6.4.2	测试系统	82
6.4.3	Benchmark	83

第 7 章 总结	85
附录 A Benchmark 汇总	86
附录 B 测试工具	87
B.1 Netperf	87
B.2 Iperf3	87
B.3 Perftest	87
B.4 Perf	88
B.5 TPC Benchmarks	88
B.6 FIO	88
B.7 Imagemagick	88
B.8 ffmpeg	88
B.9 sockperf	89
B.10 fortio	89
B.11 Ftester	89
索 引	90

插图目录

1.1 DPU 功能集	4
2.1 单端型与端到端型测试系统模型	10
2.2 单端型测试系统组成	12
2.3 端到端型测试系统组成	13
2.4 多端型测试系统模型	14
2.5 多端型测试系统组成	14
2.6 测试执行	16
3.1 DPU 加速基础网络的架构示意图	18
3.2 DPU 基础网络加速能力端到端型测试系统	19
3.3 DPU 基础网络加速多端型测试系统	20
3.4 DPU RDMA 实现内存资源池化场景示意图	23
3.5 DPU 支撑 RDMA 应用的架构示意图	23
3.6 基于 DPU 的 OVS 加速结构	27
3.7 OVS 测试系统	28
3.8 SD-WAN 应用场景	30
3.9 SD-WAN 网关节点部署	31
3.10 SD-WAN 测试系统	31
3.11 基于 DPU 的 LB 加速结构	34
3.12 LB 测试系统	35
3.13 5G MEC 中的 DPU	37
3.14 5G MEC DPU 加速结构图	38
3.15 5G MEC DPU 测试组网	38
4.1 加速结构	42
4.2 NVMe-oF 半卸载示意图	43
4.3 NVMe-oF 全卸载示意图	44
4.4 测试系统	44

5.1	基于 DPU 的 Spark 加速系统架构图	50
5.2	Spark 测试系统示意图	51
5.3	DPU AI 加速结构	53
5.4	DPU AI 加速测试系统	54
5.5	数据库加速结构	58
5.6	数据库加速测试系统	59
5.7	DPU 编解码加速结构	61
5.8	DPU 视频/图像编解码测试系统	61
5.9	DPU 加速服务网格架构	66
5.10	服务网格单端型测试系统	67
5.11	服务网格端到端型测试系统	67
6.1	DPI 加速结构	73
6.2	DPI 测试系统	73
6.3	基于 DPU 的防火墙加速架构	76
6.4	防火墙测试系统	77
6.5	IPS/IDS 加速结构	79
6.6	IPS/IDS 测试系统	79
6.7	基于 DPU 的 IPSec 隧道加速示意图	82
6.8	基于 DPU 的 IPSec 隧道测试系统	82

表格目录

4.1	大数据顺序读写负载模型	46
4.2	大数据顺序读写负载模型	46
4.3	小数据读写负载模型	47
5.1	AI 测试负载模型	56
5.2	AI 评估场景	56
5.3	vbench 视频数据集	62
5.4	服务网格延时测试负载模型	69
5.5	服务网格吞吐测试负载模型	69
5.6	服务网格应用层测试负载模型	70
A.1	Benchmark 汇总表	86

第 1 章 DPU 性能评测导论

芯片评测通行的评价维度是PPA, 即性能 (Performance)、功耗 (Power)、面积 (Area), 这三个维度可以用于比较同类芯片产品的优劣。然而, 这个评价维度适用的前提是芯片要“同类”, 例如, 基于 X86 或 ARM 指令集的服务器级 CPU; 或者即便不属于同一类指令集, 但至少是级别相近的 CPU, 并且可以运行同类操作系统。对于不同类别的芯片, PPA 的比较没有实际意义。对于新近发展起来的 DPU (Data Process Units) 芯片, 现有不同厂商的DPU 从功能角度来看, 存在较大差异。虽笼统属于 DPU 大类, 但是否属于“同类”仍有待商榷。这必然导致性能评价的维度各有侧重, 呈现多元化, 给建立一个公平的 DPU 的评价体系带来了较大的挑战。本书针对现阶段 DPU 产品的功能定义, 充分考虑 DPU 使用环境等的差异性, 试图为未来 DPU 产品建立一套公平、开放、全面、客观的 DPU 评测体系 (DPU Benchmark)。一方面为 DPU 用户提供参考, 一方面也为未来 DPU 产品的标准化提供引导。

1.1 DPU 评测的挑战和重点

要建立一套有指导意义的 DPU 评测方案, 功能定义是关键。所谓的“评测”, 难度不在于比较功耗、面积等物理参数, 而在性能的定义。在早期的同类 CPU 性能比较时, 性能的高低很大程度上反映在主频的高低上。因此在面向大众市场的宣传时往往将主频作为性能的表征参数: 主频越高, 性能越好。其实在简化这个性能比较前, 有一个前提就是二者功能是兼容的, 都可以运行相同的操作系统、相同的基准测试程序。只要在系统的访存带宽、存储速度等因素都无差异的前提下, 以主频表征性能的做法准确度就很高。然而, 由于 DPU 不具备 CPU 同类的前提, 所以 DPU 如果也用主频作为性能的表征参数, 结论则会非常不可靠。鉴于 DPU 目前在网络场景下的应用较多, 所以也有业界人士把以太网物理带宽作为表征 DPU 性能指标的参数。例如 Marvell 的 OCTEON10 DPU 产品, 标称 400Gbps; Nvidia 的 BludeField 2 代和 3 代产品, 分别标称 200Gbps 和 400Gbps。这项指标对于单纯的网卡、交换芯片具有一定的代表性; 但是对于 DPU 而言, 显然不足以反映 DPU 的性能优劣, 否则各家厂商也就不必在 DPU 结构设计和优化上做大量的努力和创新了。

DPU 在以“软件定义”为核心技术路线的大背景下, 支持各种软件定义技术, 如软

件定义网络 (Software Defined Network, SDN)、软件定义存储 (Software Defined Storage, SDS)、软件定义加速器 (Software Defined Accelerator, SDA) 等, 成为 DPU 的一个重要特征。在这些软件定义技术框架下, 通常都会采用控制与数据分离的架构。典型的如 SDN 中有两个基本逻辑层次的定义, 基本决定了 SDN 整体的技术架构: 1) 数据平面 (Data Plane), 定义是用于数据包解析和处理的数据通路, 代表计算与数据密集的功能部分; 2) 控制平面 (Control Plane), 定义是为输入输出数据流提供和配置数据平面的算法集合, 代表资源调度、系统配置、链路建立等控制密集的功能部分。此外, 业界通常还增加了第三个层面, 即 3) 管理平面 (Mgmt. Plane)¹, 代表系统监控、故障隔离、在线修复等周期性或偶发性的部分应用。如果把一个城市的路网基础设施比做 SDN, 那么纵横交错的道路就是其“数据平面”, 其路网密度和宽度决定了路网的流量上限; 所有的交通灯及其控制系统就是其“控制平面”, 其控制算法优劣、部署位置的合理程度决定了交通流量实际容量; 各种测速点、流量监控、临时交通管制、事故拥塞疏导等就是其“管理平面”。有了这套基础设施, 各类用户就可以使用各种车辆 (相当于用户的应用程序) 来开展运输服务了。

对于一个分布式的计算系统, 这个划分是按照逻辑功能的, 并不意味着在物理系统上需要使用不同硬件设备; 相反, 在实际系统中, 出于灵活性、部署成本等因素考虑, 即使性能上没有优势, 用于执行数据平面算法和控制平面算法的可能都是通用服务器。DPU 的引入能够提供更好的解决方案。数据平面的性能提升可以更多依赖 DPU 中的专用硬件、专用处理单元; 控制平面的开销降低可以通过将控制算法部分卸载到 DPU 端来实现, 甚至运行完整的管理程序 (Hypervisor), 以降低 CPU 占用率, 将更多的通用算力释放给上层应用; 管理平面的灵活性可以通过 DPU 支持的各种遥测功能得到强化。通常, 数据平面任务的性能是计算系统性能的决定性因素, 决定了系统性能的理论峰值水平; 控制平面的性能决定了系统对高并发度的支撑, 是提供灵活性的关键; 管理平面决定了系统的可用性、故障诊断、应急处理以及运维等的方便程度。因此, 从性能的视角看, 在评测过程中, 数据平面的性能量化是首要的, 其次的控制平面; 对于管理平面的评价通常只需要用是否“支持”某种管理协议, 是否提供某种管理接口即可。一种理想的 DPU 对于系统的作用不仅仅是加速几个数据面的算法, 卸载几个控制面的算法; 更重要的是与 CPU 构成一个新的计算平台, 使之可以更高效地支撑已有的应用, 并为未来的新应用提供必要的运行基础。因此, DPU 的评测首先要能突出 DPU 在基础平台层面的作用。

¹还有一种说法是增加第四个平面: “应用平面”, 作为支撑上层应用的接口集合, 但由于这通常不是性能的影响因素, 因此在此没有列出。

所谓的“基础层”是指由网络、存储、计算（即服务器）等硬件资源和其上运行的虚拟机、操作系统构成的可支撑上层应用运行、开发、运维等操作的资源平台。简言之，基础层就是网络、存储和计算构成的资源池，既可以是物理资源（物理机、包括服务器、网络、存储设备等），也可以是逻辑资源（虚拟机）。然而，这些基础功能在现有体系下是紧密耦合的。例如，网络协议的处理依赖服务器中的 CPU 资源，不同虚拟机之间的存储转发也由运行在主机端的软件实现的交换机（OVS）来处理，因此也依赖于主机端的 CPU 资源；分布式存储引擎传输层也依赖于网络，同时存储中的缓存、加密、去冗余、压缩、纠删码和 RAID 等也要依赖于 CPU；甚至最常见的 TCP/IP 协议栈的处理，也依赖于操作系统中的软件协议栈。这样不仅要消耗大量的计算资源，更重要的是影响系统的性能！网络协议处理时延，网络数据频繁在不同系统态下的拷贝，频繁主机中断的性能开销等都成为了潜在的系统性能瓶颈。

1.2 开展 DPU 评测建议标准

DPU 可以实现网络、存储和计算在一定程度的解耦，而解耦可以为专用化提升性能和效率创造条件。例如，将部分网络协议处理的功能卸载到 DPU 网卡，计算（服务器）不再分资源来处理这些网络数据协议，从而实现一定程度的网络与计算解耦；把存储中的缓存、加密、去冗余、压缩、纠删码和 RAID 等功能从主机端的 CPU 卸载到发起（Initiator）和目标（Target）侧的 DPU，从而实现存储与计算的解耦。

在开展性能评测的过程中，可以基于为网络、存储和计算分别构造的测试系统来测试 DPU 对系统的影响。对于构造的测试系统有三个方面的建议标准，简称为“POC 原则”：

第一、真实性（Practicality）原则，即测试系统能反映实际应用场景。这就需要测试用例不仅要反映最常用的场景，还要覆盖各种典型的长尾场景（Corner Cases），并且充分利用现有的一些基本测试工具和一些成熟的基准程序，用好现有的“轮子”。

第二、针对性（Orientation）原则，即测试系统要能突出特定方面的性能表现，避免因其它不相关的性能瓶颈导致测试激励无法发挥作用。测试目标有针对性才便于设计有针对性的测试基准程序和数据集。同时，由于不同用户场景的需求不同，针对性原则更利于用户做出合理的选择。

第三、全面性（Comprehensiveness）原则，从 DPU 的功能角度出发，应该包括网络、存储、计算三个维度的测试，并且还可以构建一定的“网络 + 存储”、“网络 + 计

算”等组合测试，更全面反映 DPU 的应用效果。

基于以上的“POC”原则，本技术白皮书的内容试图为 DPU 的性能评测建立一套系统的评测参考框架，涉及测试系统的构建，测试负载的刻画，测试指标的选取，测试用例的设计等关键问题。

1.3 系统测试的四个维度

DPU 系统评测可以应用于多种场景，可归因于图1.1所示的四个维度：

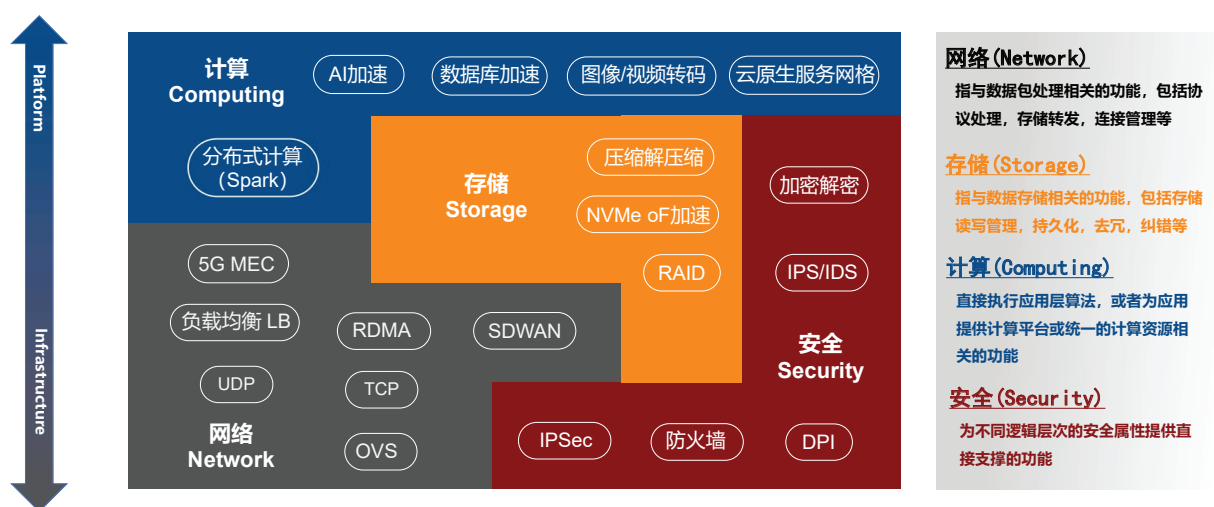


图 1.1: DPU 功能集

1.3.1 网络维度

网络维度指与数据包处理相关的功能，包括协议处理，存储转发，连接管理等功能集合。网络场景是 DPU 发挥的“主战场”之一，包括传统的 TCP/IP 网络，低延迟、高带宽网络，也包括网络虚拟化等。网络卸载加速能力是 DPU 上各种业务和算力的重要基础，直接影响其它多种业务和算力性能的上限，尤其是在以高性能网络互连而成的融合异构算力的高性能分布式计算系统中。DPU 实现网络加速卸载的主要场景涉及云计算平台 VPC 网络卸载（如 Open vSwitch 加速卸载，并开启或关闭 Connection Tracking 和 INT），协议栈加速卸载（如 TCP/IP Offloading，RDMA 网络等），云原生网络加速卸载（如 OVN-Kubernetes）和其它 NFV 网络网元（如 SD-WAN）等应用场景。在各应用场景下的主要评测维度包括：

- 表项（会话）规模，如流表规模、路由表规模、安全组 ACL 规模和会话连接的规模（如 TCP 连接，RDMA 的 QP）等，用来表达 DPU 网络支撑节点规模能力；

- 吞吐能力，如 DPU 的转发能力PPS（Packets Per Second，包每秒）和BPS（Bits Per Second，bit 每秒），用来表达 DPU 网络数据包的转发处理能力；
- 建表性能，如流表的创建能力CPS（Connection Per Second，连接每秒）和路由表的创建能力（如路由条目每秒），用来表达 DPU 网络支持并发会话或路由学习的能力；
- 时延性能，如数据包经过 DPU 转发所消耗的时间，或者数据经过 DPU 处理所消耗的时间，用来表达 DPU 网络转发或处理数据包的效率；
- 网络可观测能力，即 DPU 上基于物理端口、虚拟机实例、vNIC 和数据流级别的可观测能力，用来表达 DPU 网络的统计和遥测能力；
- 确定性网络能力，如保证数据包传输的网络QoS 能力（即物理端口、虚拟机实例、vNIC 和数据流级别的限速、带宽保障和调度能力），低丢包率或零丢包率能力，有界低延迟能力（即有限的长尾延迟问题），网络的低抖动能力（即低数据包延迟变化），用来表达 DPU 网络的可靠数据传输能力。

1.3.2 存储维度

存储维度指与数据存储相关的功能，包括存储读写管理、持久化、去冗、纠错等功能集合。存储 I/O 通路是指计算单元与本地或远程的数据存储器与之间的通路。传统的存储 I/O 是将持久化存储的数据调入内存参与计算，或者将数据存入存储器并持久化保存，通常通过 DMA 机制来完成。DPU 上存储加速卸载功能主要是用于连接算力与存储服务，为高性能计算中存算分离场景提供算力和存储的独立弹性伸缩能力及高性能并行存储 I/O 功能。在“数据中心税（Datacenter tax）”中，涉及存储相关的计算量大约占数据中心服务器运行周期的 18%~23%。随着非易失存储介质（NVM）访问延时数量级的降低，以及数据中心高速互联网络的部署，存储性能的瓶颈已经转移到服务器 CPU 的处理能力。由此，“拓宽”存储的通路就尤为重要。DPU 可以从两方面加速存储性能：首先，通过将主机端的网络协议栈卸载到 DPU 中，主机端不需管理复杂的网络细节，只需要发起读写命令；其次，通过实现存储资源的硬件虚拟化，主机中运行的虚拟机或者容器可旁路管理程序（Hypervisor）直接访问外设的存储资源。

DPU 上存储加速卸载的主要场景是NVMe-oF，包括 NVMe/RoCEv2 和 NVMe/TCP 等协议，结合 NVMe-oF 的存储服务加速卸载能力，如 RAID、纠删码、加解密和压缩解压缩等。DPU 上存储加速卸载在 NVMe-oF 场景下的主要评测维度包括：

- NVMe-oF 存储的 I/O 读写能力，如IOPS（Input/Output Operations Per Second，I/O

操作每秒)，用来表达 DPU 存储的读写能力；

- NVMe-oF 存储的吞吐能力，如 MBps (Megabytes Per Second, 兆字节每秒)，这个能力与 DPU 主机接口和网络接口硬件规格强相关，是另一种表达 DPU 存储的数据读写能力的方式；
- NVMe-oF 存储的时延性能，如块数据写入 NVMe-oF 远端磁盘或从远端 NVMe-oF 磁盘读出块数据所需要的时间，用来表达 DPU 存储的数据读写和传输的效率；
- NVMe-oF 存储的 QoS 能力，如卷级别和虚拟机实例级别的 IOPS 或 MBps 的限速、带宽保障和调度能力，用来表达 DPU 存储的可靠 I/O 读写能力。

1.3.3 计算维度

计算维度指直接执行应用层算法，或者为应用提供计算平台或统一计算资源的相关功能集合。数据中心中不仅有 CPU 集群，也有 GPU 集群（并行计算）、FPGA 集群（定制计算），AI 加速器集群（智能计算）。这些加速器集群用于加速不同类型负载，以获得比 CPU 更高的加速比。一方面，DPU 同样具备异构计算的特征，并且可以作为多种专用加速器的载体，可用于加速数据中心的一些典型应用，如数据库查询、视频图像转码等，未来也可能用于 AI 计算，类似于 Marvell 的 Octeon10 DPU 中的 AI 功能。另一方面，DPU 也能通过卸载一些计算平台的服务来构建更高效的计算平台。例如，随着云原生微服务架构的普及，服务网格作为微服务间通信的专用基础设施层，服务网格数据面代理的边车（sidecar）功能可通过 DPU 实现服务网格数据面代理，进而降低 CPU 和内存的消耗以及微服务间的访问延迟。

DPU 上计算加速卸载的功能主要包括大数据计算加速、ML/AI 算法加速和视频编解码等。通过 DPU 上硬件实现的大数据计算算子、ML/AI 算法和视频编解码等专用电路来加速云计算平台下的高性能计算业务。DPU 上各计算加速卸载场景的主要评测维度包括：

- 计算业务的吞吐能力，如视频编解码处理时的吞吐量，用来表达 DPU 上计算业务的处理能力；
- 计算业务的时延性能，如大数据计算、ML/AI 计算等执行一个任务所消耗的时间，用来表达 DPU 上计算业务的处理效率；
- 计算业务的并行处理能力，如大数据计算、ML/AI 计算和视频编解码等业务在 DPU 上并行处理的任务和会话数量，用来表达 DPU 上计算业务的并行处理能力；

- 计算业务的能效比，如计算单位数据所消耗的功耗，用来表达 DPU 上计算业务的处理效能。

1.3.4 安全维度

安全维度指为不同逻辑层次的安全属性提供直接支撑的功能集合。通常，安全是系统希望具备的一种重要属性，而不是某个物理实体。这有别与网络、存储、计算等物理“实体”。但是鉴于安全在确保系统的可用性、可信性方面的重要意义，所以也建议作为相对独立的一个场景来进行评测。安全是业界研究的一个热点，涉及网络安全、内容安全、硬件安全、软件安全等不同维度和逻辑层次，不太可能用一种场景下的安全来代表“安全”。考虑到加密解密是很多安全机制的基础，所以我们将加密解密算法的性能评估作为其中一项重点。同时，目前在网络安全中较为成熟的互联网安全协议（Internet Protocol Security, IPSec）（通过对 IP 协议的分组进行加密和认证来保护 IP 协议的网络传输协议族），由于其具有一定的代表性，所以我们也作为评测的一个重点内容。

DPU 对于安全属性的增强主要体现在提供 DPU 上网络、存储和计算等业务的安全支持，主要包括硬件信任根、加解密算法以及三元匹配和正则匹配等匹配查找算法等。DPU 上安全加速卸载的主要应用场景涉及证书密钥交换、传输中数据加解密、静态数据加解密以及基于三元匹配和正则匹配的安全应用加速卸载等。以网络安全为例，DPI、防火墙可以做到一定的安全，但通过软件实现的安全已经占用大量的 CPU 处理时间，影响应用程序的性能以及用户体验，而 DPU 本身作为网络的基础设施，具备实现“硬件防火墙”的能力，尤其是安全中计算模式固定的加解密算法可以采用 DPU 内置的加解密引擎。DPU 上各安全加速卸载场景下的主要评测维度包括：

- 安全策略规模，如下发到 DPU 上的安全策略条目，用来表达 DPU 上支持的安全策略规模；
- 安全业务的吞吐能力，如加解密吞吐能力和安全应用匹配查询的吞吐能力，用来表达 DPU 上安全加解密处理和安全过滤匹配查询的能力；
- 安全业务的时延性能，如加解密业务或安全应用匹配查询消耗的时间，用来表达 DPU 上安全加解密处理和安全过滤匹配查询的效率和效能；
- 安全业务的并发会话规模，如加解密业务并行会话数量或安全应用并行匹配查询的会话数量，用来表达 DPU 上安全加解密处理和安全过滤匹配查询的并行处理能力。

1.4 内容的组织结构

本技术白皮书后续的内容安排如下：

第二章定义了三种 DPU测试系统（System Under Test, SUT）模型：单端型（Single-End）测试系统，端到端型（End-to-End）测试系统和 多端型（Multi-End）测试系统；并给出系统原型和适用场景。同时，规定了每种基准测试遵循执行的流程，旨在最大程度确保结果的有效性和可复现。

第三、四、五、六章分别从网络、存储、计算、安全四个系统测试维度介绍了 DPU 在对应场景中的基本架构，并给出实际测试系统模型。最后，根据不同的应用场景提出评测方案。

第七章总结全文。

第 2 章 DPU 性能评测系统框架与测试流程

本章介绍 DPU 性能测试系统框架与测试流程，包括测试系统、测试要求、测试活动三部分。具体的，测试系统定义了三种搭建 DPU 测试系统的方法，测试要求阐述了组建 DPU 测试系统时需要满足的要求，测试活动定义了 DPU Benchmark 的选择策略、执行前准备、执行过程及测试结果报告。

2.1 DPU 测试系统 (SUT)

DPU 测试系统 (System Under Test, SUT) 是测试 DPU 性能的平台，其结构在不同应用场景中，主要分为三类：单端型测试系统 (Single-End)、端到端型测试系统 (End-to-End) 和多端型测试系统 (Multi-End)。本白皮书中描述的所有 Benchmark 测试系统都可以归类为这三种基本类型。

单端型 (Single-End) 测试系统是主机与 DPU 通过总线互联构成的封闭测试系统，主要用于模拟无网络连接情况下，DPU 作为专用加速器执行主机端特定业务的过程。在这种系统中，计算与数据传输仅在主机与 DPU 之间进行，DPU 不与其他设备通信，仅执行特定计算业务。单端型测试系统主要测试 DPU 对特定计算任务的性能提升。

端到端型 (End-to-End) 测试系统将两个单端型测试系统通过简单网络相连 (网线直连)，其中一个为请求发起者 (Initiator/Client)，另一个为请求接受者 (Target/Server)。端到端型测试系统主要模拟 DPU 作为网络加速器执行双端操作的场景。在这种系统中，计算与数据传输不仅在主机与 DPU 之间进行，还通过网络传输到其他系统。DPU 在 Server 端接收主机端发起的网络请求并通过网口转发，或者在 Client 端接收网络的请求并向作出响应。端到端型测试系统能够屏蔽网络性能 (转发次数、交换设备的性能、转发设备的性能) 对 DPU 性能的影响，主要测试 DPU 对网络协议卸载、网络数据包处理的峰值能力。

多端型 (Multi-End) 测试系统是多个单端型测试系统通过复杂的网络拓扑相连的开放测试系统。该测试系统主要模拟 DPU 作为网络加速器在复杂环境中处理网络任务的场景。在这种系统中，网络环境 (网络拓扑) 与资源分布 (相同或者不同类型的 DPU 系统) 情况复杂，DPU 不仅承担一对一的网络任务，还可能接收一/多个其他系统的请求 (含背景流量甚至恶意攻击) 或响应一/多个其他系统的请求。多端型测试系统主要测试

DPU 的实际网络处理能力。

2.1.1 单端型 (Single-End) 测试系统

2.1.1.1 测试系统定义

主机与 DPU 通过总线（通常为 PCIe）直接相连形成的可独立工作的系统为单端型测试系统，系统包含支持 DPU 正常工作的软件和硬件。在这种测试系统中，DPU 作为专用加速器执行特点计算业务。

2.1.1.2 测试系统模型

单端型测试系统模型的基本结构如图2.1(a)所示。运行时环境 (Run Time Environment, RTE) 代表运行中的 DPU 测试程序，交换设备 (Switch) 为 DPU 与主机系统 (Host System) 的互连结构（通常为 PCIe）。在单端型测试系统中，DPU 作为专用加速器，主机使用 DPU 加速某些任务的处理，如数据库查询、AI 训练等。任务数据首先从主机端搬运到 DPU，DPU 计算完成后，再将处理完的数据写回主机。在这种工作模式中，DPU 测试程序可以以数据处理时间、数据处理规模作为性能指标。

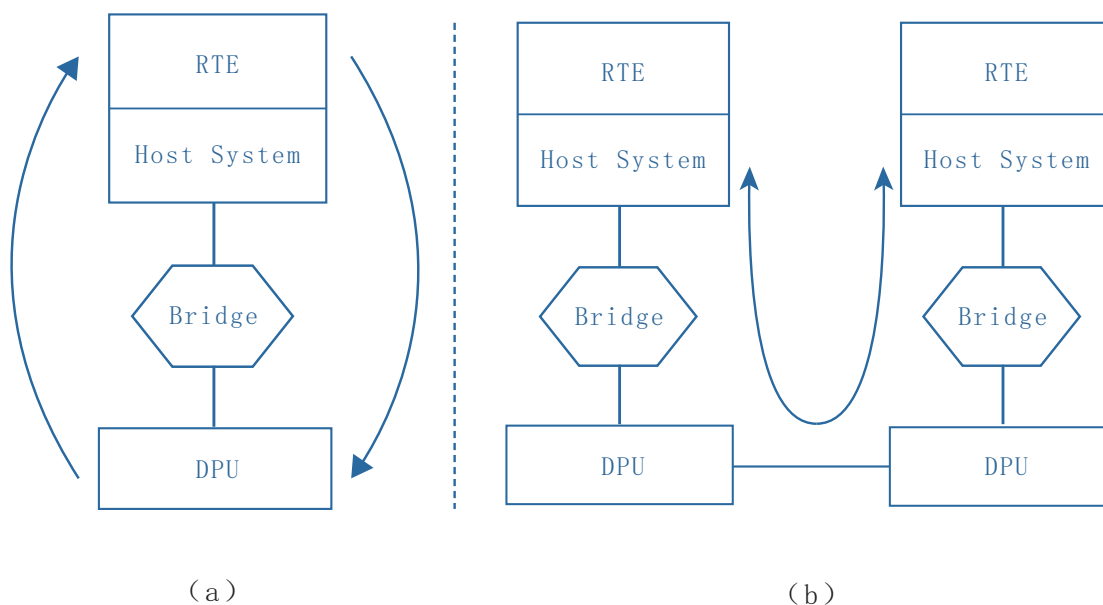


图 2.1: 单端型与端到端型测试系统模型

2.1.1.3 测试系统组成

单端型测试系统具体由硬件和软件两部分构成，硬件主要有 DPU、CPU 以及其他保证系统正常进行的硬件（内存、硬盘等），软件主要有操作系统、DPU 驱动、开源工具、DPU Benchmark，结构示意图如图2.2。各部分组件要求如下：

1. DPU Benchmark，评测 DPU 系统性能的基准测试程序；
2. 开源工具（Open Source Tools），一些广泛使用的性能测试工具，保证测试结果可重复，在不同的用户测试时可以获得相近的结果；
3. DPU Benchmark 驱动（Benchmark Driver），基于内核态的测试驱动或者基于用户态的测试驱动；
4. DPU 驱动（DPU Driver），包括 DPU 最底层的抽象，也包括 DPU 所提供给用户程序使用的接口；
5. 操作系统（Operating System），需要支持 DPU Benchmark 的版本，且相关信息需要在结果报告中明确写出，如操作系统版本，内核版本等；
6. 计算设备（Computing Device），通用 CPU，其上运行操作系统；
7. 连接设备（Connected Device），根据 DPU 板卡接口不同选择不同，常见的有 PCIe、CXL；
8. IO 硬件（I/O），鼠标、键盘等保证系统执行所必须的硬件设备；
9. 被测 DPU 产品（DPU）；
10. 其他硬件资源（Hardware Resources），如内存、硬盘。

2.1.2 端到端型（End-to-End）测试系统

2.1.2.1 测试系统定义

端到端型测试系统是由两个单端型测试系统通过网线直连的方式组成的测试系统。在这种测试系统中，屏蔽了复杂的网络拓扑与其他系统流量的影响，可以测试 DPU 网络加速的峰值性能。

2.1.2.2 测试系统模型

端到端型测试系统由两个单端型测试模型直接相连（如图2.1(b)所示），每个单端测试系统分别运行 DPU Benchmark 测试程序，一般其中一个运行发起者程序 (Initiator/-

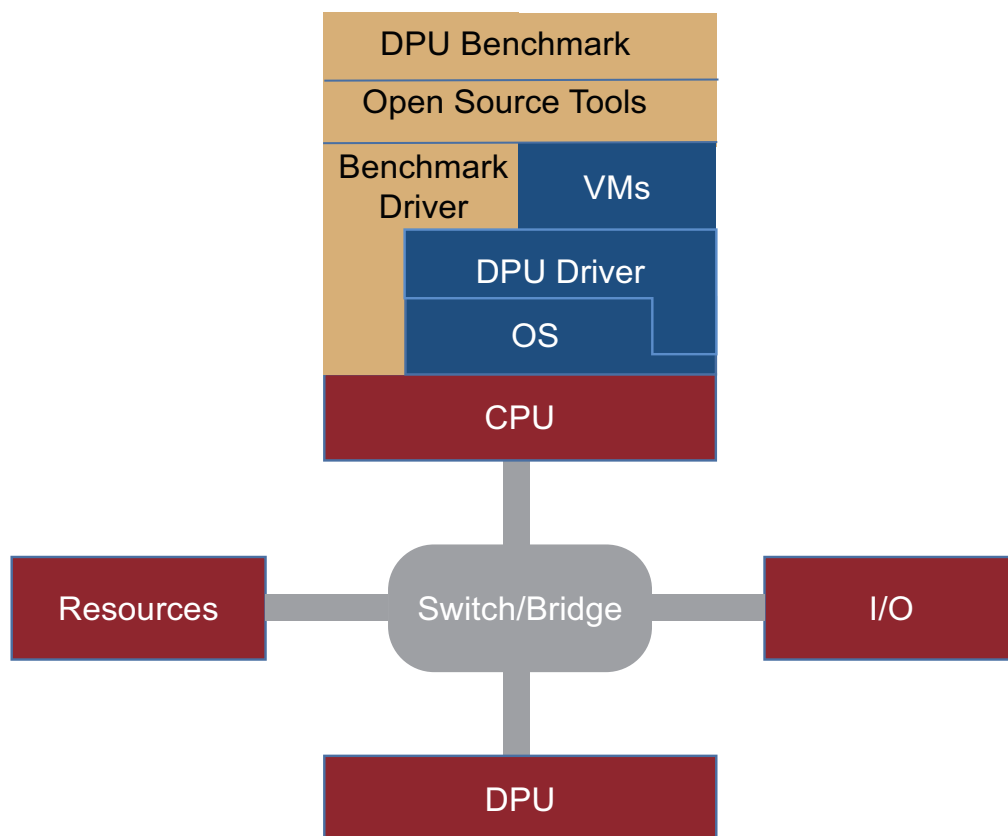


图 2.2: 单端型测试系统组成

Client), 另一个运行接收者程序 (Target/Server)。通过发起者发送请求, 接收者响应请求, 测量基于网络的端到端操作, 例如 RDMA Send/Receive 操作, NVMe-oF 远程访问。在这种测试模型中, 根据从一端 DPU 系统到另一端 DPU 系统的请求时间和流量等测试 DPU 的延迟、吞吐量等性能。

2.1.2.3 测试系统组成

端对端型测试系统在单端型测试系统的基础上, 增加网络基础设施 (通常为网线) 用于直接连接两个单端测试系统, 其结构如图2.3所示。根据 DPU 所支持的网口速率不同, 可采取不同的网络连接设备, 缺省值为 100Gbs 连接。

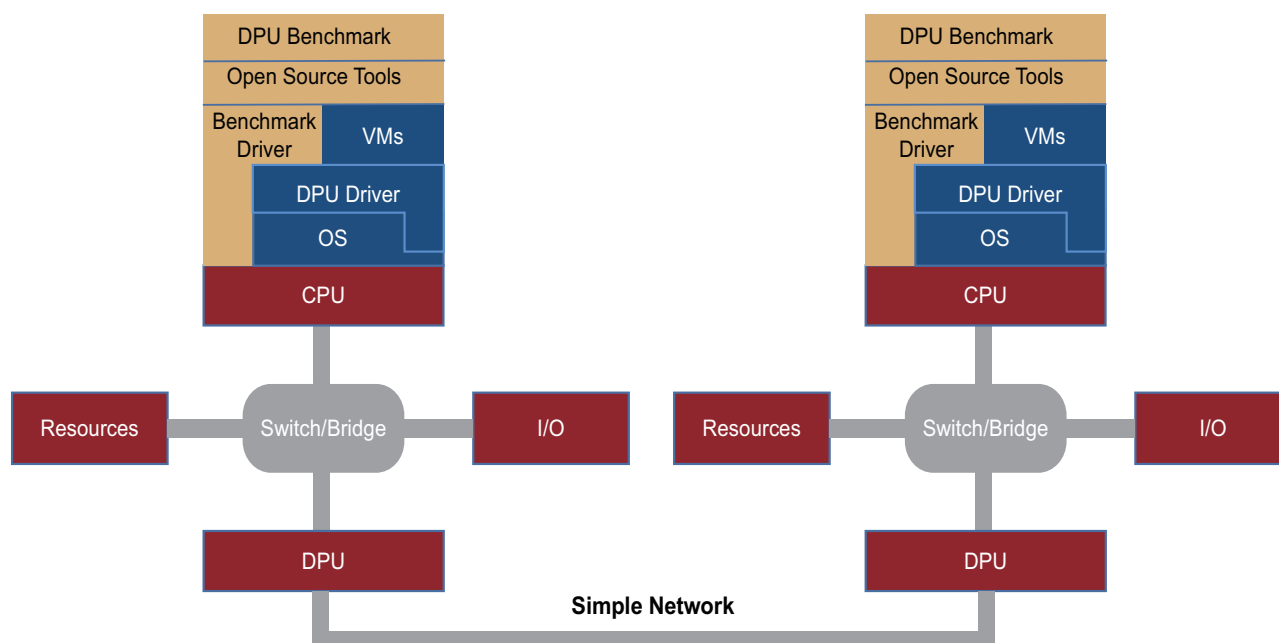


图 2.3: 端到端型测试系统组成

2.1.3 多端型 (Multi-End) 测试系统

2.1.3.1 测试系统定义

多端型测试系统是由多个单端型测试系统经由复杂网络（复杂的网络拓扑）连接形成的测试系统。在这种测试系统中，可以模拟 DPU 所处的真实网络环境，测试 DPU 网络处理的实际性能。

2.1.3.2 测试系统模型

多端型测试系统模型主要用于模拟基于 DPU 构建的数据中心，通过复杂的网络将众多 DPU 系统（功能相同或者不同）互连起来，从而测试基于 DPU 系统实现的网络、存储或安全加速应用的真实性能，如 NVMe-oF 读取不同结点存储系统的存储资源、RDMA 访问不同结点的资源。在这种测试模型中，根据请求与响应时间、请求的数据量及并发的请求数等可以测试延时、吞吐量、连接能力等指标。

2.1.3.3 测试系统组成

多端型测试系统在单端型测试系统的基础上，需要增加更复杂的网络基础设施用于连接不同节点的 DPU 系统。这些网络基础设施包括但不限于集线器、交换机、路由

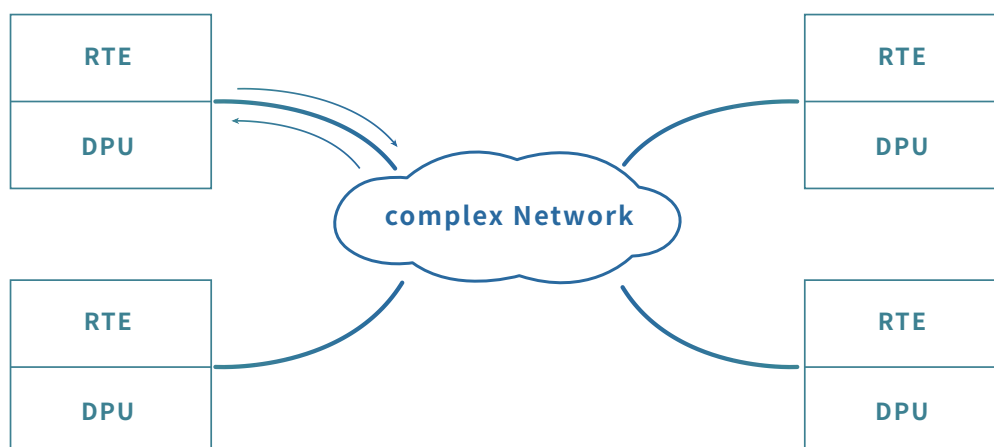


图 2.4: 多端型测试系统模型

器等。如图2.5

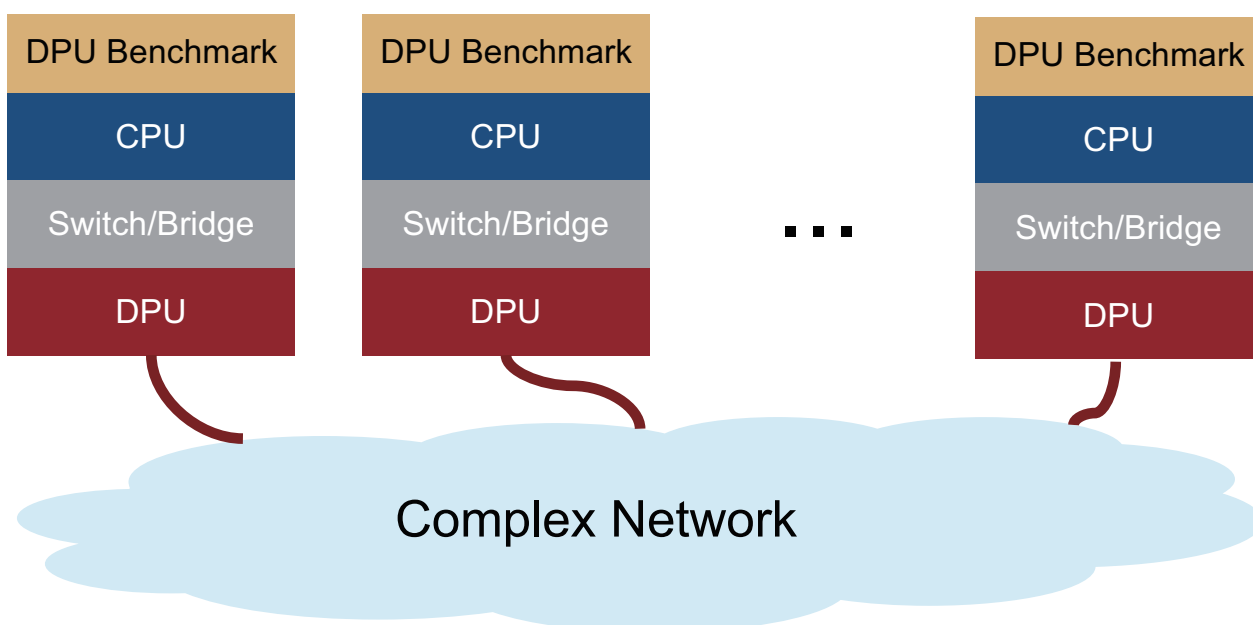


图 2.5: 多端型测试系统组成

2.2 测试要求

利用 DPU Benchmark 进行 DPU 性能测试时需要满足一定的测试要求：

- 执行 DPU Benchmark 时必须严格按照 DPU 测试系统搭建，如果对测试系统有改变需要补充说明；
- 选择基准测试程序（benchmark）时，需要选择相关章节中推荐的测试工具及基准程序；

- 测试同一款 DPU 产品时，不同用例的 DPU 测试系统软硬件配置应保持一致，以保证测试结果的公平。

2.3 测试活动

测试活动描述了采用 DPU Benchmark 测试 DPU 系统需要执行的一般步骤。

2.3.1 测试选择

DPU Benchmark 覆盖了 DPU 所有的应用场景。在测试 DPU 时，可以根据其具备的功能选择相应的基准程序进行测试。

2.3.2 测试准备

执行 DPU Benchmark 相关测试时，需要根据不同的应用场景搭建 DPU 测试系统。系统搭建完成后，安装需要使用的测试负载生成器及 DPU 配套的相关软件（驱动、应用程序）。

2.3.3 测试执行

在 DPU Benchmark 执行时，为了保证测试结果的准确性与可重复性，对于同一种测试用例执行两轮（特殊的情况在具体的测试用例说明）。每一轮分为三个部分：预热 (Warmup Run) 部分排除初始化干扰；负载执行 (Measured Run) 部分为真正的指标测试；结果检查 (Result Check) 部分检查结果的正确性 (数据传输的完整性，计算结果的正确性等)。每执行完一轮需要恢复至初始状态 (Clear up)。在预热部分与负载执行之间不允许执行其他任务。每一轮执行的过程中，DPU 测试系统配置不可以发生变化。DPU Benchmark 执行阶段如图2.6所示：

在 DPU Benchmark 执行过程中，DPU 测试系统不可以重启或者重新开始。如果 DPU 系统测试应用、操作系统或者硬件发生不可恢复的错误时，这一轮的执行应作无效；如果 DPU 系统测试应用、操作系统或者硬件检测到可恢复的错误，并且错误纠正并恢复正常，那么这一轮的执行应作有效。在错误发生处理期间，测试者不能人工干预，否则，这一轮执行应作无效。



图 2.6: 测试执行

2.3.4 测试报告

2.3.4.1 性能报告

根据 DPU 性能测试过程中实际执行的 DPU Benchmark 给出评价某一类计算负载的每个基准程序的测试结果。测试结果一般包括：所选测试基准程序，性能指标测量值。

对于没有执行的基准程序需要明确标注，如，该 DPU 不支持此功能。

2.3.4.2 参数报告

测试报告中除了给出不同测试用例的性能指标，还需要给出测试过程中的全部配置信息，包括：

- 配置参数，包括服务器、存储、网络等其他硬件设施；
- 安装的操作系统、文件系统、测试工具等软件版本；
- 在测试过程中使用到的其他软件程序；
- 程序编译优化选项。

在执行 DPU Benchmark 时如果系统发送异常或者其他情况，需要在报告附录中指出。

第3章 面向网络的基准评测

在现代数据中心中，软件定义网络的技术普遍使用，每年出货的大约 3000 万台服务器中有三分之一用于运行数据中心的软件定义网络协议栈¹，这种工作负载的增长速度甚至已经超越了 CPU 性能增长的摩尔定律，如果没有新的技术来应对这一情况，数据中心中用于运行真正业务层负载的 CPU 比例将越来越少，DPU 通过卸载软件定义网络的相关技术实现加速功能。不仅是数据中心，DPU 在核心网、边缘网、接入网等各个环节都可以加速网络协议处理。例如在电信网中的 SD-WAN 是构建低成本的私有广域网的技术，其中就需要的软件定义网络和高容量的数据转发支持；在 5G 接入网场景下，通过 5G 连接的终端设备通过 UPF 网关来接入 MEC 计算中心，导致了 UPF 网关成为了流量的瓶颈，也可以通过 DPU 来解决。DPU Benchmark 网络部分从如下网络加速应用作为基准测试，首先是对基础网络协议的加速，如 TCP/IP、UDP/IP，RDMA 等，降低 CPU 请求网络数据的延迟；其次是对软件定义网络技术的加速，如 OVS、SD-WAN；然后是对网络质量的管理，如负载均衡；5G MEC 也作为解决 UPF 网关流量瓶颈地重要技术作为测试基准的一部分。DPU Benchmark 网络部分主要表征 DPU 的对网络的加速能力。

3.1 基础网络加速

在数据中心基础设施中，数据中心网络是连接服务器进行大规模分布式计算的桥梁，提供实现各类资源聚合的关键路径。为支持云计算和数据量的大规模扩展，数据中心基础设施需要持续优化数据中心网络的传输质量，提高服务器处理网络数据。数据中心网络传输质量的影响因素包括网络拓扑，底层协议类型，以及交换机、路由器等网络硬件的参数配置。对于网络传输质量的评估方法和指标已在互联网工程任务组（IETF）出版的数据中心基准测试方法²被明确定义，与其不同，DPU 的基础网络基准测试重点关注服务器端，衡量 DPU 的基础网络加速能力。

传统的服务器网卡一般只处理二层以太网数据帧的封装和解封装，所有网络数据包最终都需要在操作系统对应的网络协议栈由 CPU 统一处理。随着数据中心网络带宽

¹<https://blogs.nvidia.com/blog/2021/10/29>

²详情参考 [RFC 8239]

的高速增长，CPU 的处理速率已经无法满足 QoS 要求。DPU 通过硬件卸载协议栈，减轻了主机 CPU 处理网络协议栈的资源占用，通过定制设计实现网络加速，提升服务器的网络处理能力以匹配现有高速网络带宽。

3.1.1 基于 DPU 的基础网络加速结构

图3.1给出了一种 DPU 加速基础网络的架构示意图，其中 DPU 实现 TCP/UDP/IP 协议栈的全卸载。通过为应用层提供标准 socket 网络接口，DPU 可以支持上层应用对现有的网络数据包处理业务逻辑的透明卸载。DPU 实现 TCP/IP 协议栈的网络数据处理功能的硬件卸载，通过高度定制的网络包处理功能单元，获得更低的延迟和更高的带宽同时，释放了服务器 CPU 处理 TCP/IP 协议时的占用资源，从而减轻服务器压力。

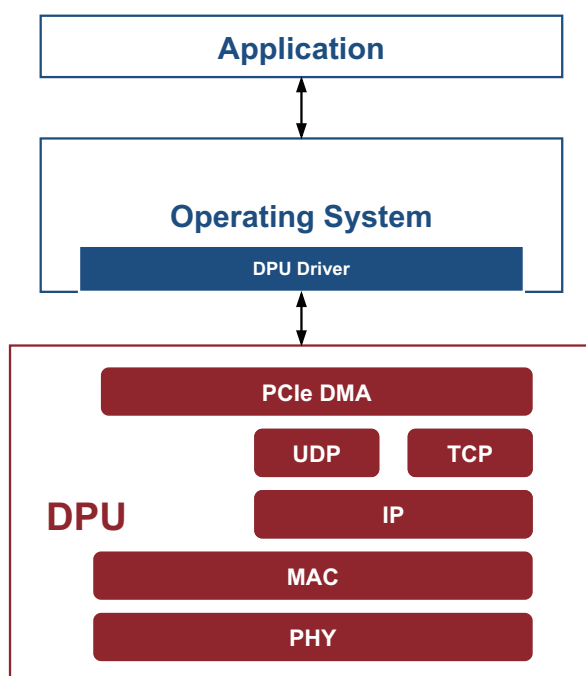


图 3.1: DPU 加速基础网络的架构示意图

3.1.2 测试系统

3.1.2.1 端到端型测试系统

基础网络端到端型测试系统适用于延迟指标的评估，如图3.2所示，采用两个 DPU 通过 100Gbps 线缆直连的网络拓扑，不引入交换机，不涉及网络拥塞，避免了网络质量变化引入的误差。因为网络拓扑中只有线缆引入额外的链路传输延迟，因此端到端测试

系统适合评估待测 DPU 峰值数据包处理性能下的端到端往返延迟。端到端往返延迟定义为，从 client 侧流量生成器发出请求数据包到 client 接收到 server 返回的响应数据包，总共经历的时延。端到端型测试系统采用对称设计，两端配置相同 DPU，服务器及其他配置规格保持一致，从而往返延迟可以真实反出 DPU 在接收和发送数据包处理的理想延迟。

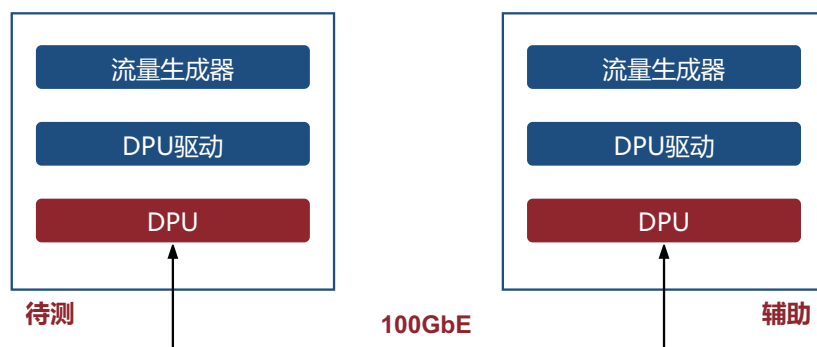


图 3.2: DPU 基础网络加速能力端到端型测试系统

3.1.2.2 多端型测试系统

基础网络多端型测试系统适用于 DPU 可达带宽和包率的评估，如图3.3所示，采用一个 DPU 通过交换机和多个 DPU 互联的拓扑。多端型测试系统适合评估待测 DPU 接收或发送方向的最大带宽，以及支持 TCP 并发链接的能力。根据待测流量方向不同，可以更加具体地划分为一对多测试系统和多对一测试系统。

一对多测试系统中，待测 DPU 作为 client 侧，多个辅助 DPU 作为 server 侧。Server 侧每个 DPU 具有独立的 IP 地址，client 侧流量生成器通过多进程的执行方式向不同的 server 侧 DPU 输出流量。通过将待测 DPU 发出的流量均分到多个 server 侧 DPU，避免了 server 侧 DPU 接收侧性能不足对发送侧性能评估的干扰。

多对一测试系统中，待测 DPU 作为 server，多个辅助 DPU 作为 client。Server 上 DPU 同时开启多个进程，每个进程监听不同 DPU client 的连接端口。DPU client 在同一时间向待测 DPU server 输出流量。

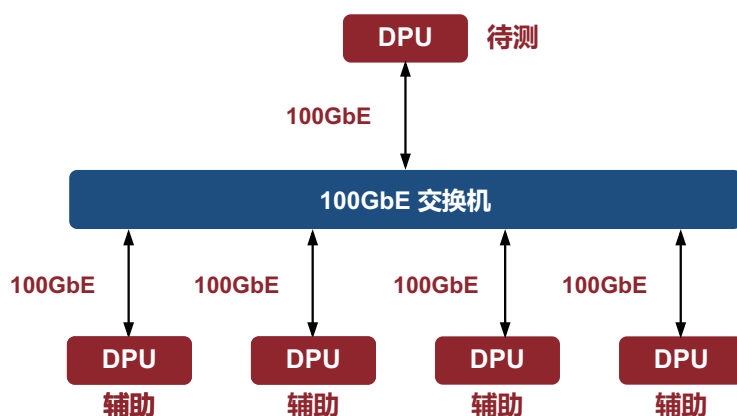


图 3.3: DPU 基础网络加速多端型测试系统

3.1.3 Benchmark

3.1.3.1 测试负载参数

传输层协议类型 (transmission protocol layer) 传输层协议类型分为 TCP 与 UDP。

测试时间 (test time) 单次测试执行时间，单位为 s。

消息字节数 (message bytes) 测试过程中单个消息的大小，单位为 Byte。

带宽 (bandwidth) 单位时间内传送的比特数，单位为 bps。

并发连接数 (concurrent connection) 客户端 (Client) 和服务器 (Server) 之间建立的活跃 TCP 链接数目。

3.1.3.2 测试负载模型

端到端延迟测试模型 互联网上大部分流量属于短流，短流的消息字节数小。短流的特点是对延迟较为敏感，且通常具有响应时间要求，一般出现在网页浏览、即时通讯等业务。端到端延迟测试负载模型中，规定消息字节数小于链路最大传输单元来模拟短流，侧重于评估待测 DPU 对短流的处理延迟。

带宽测试模型 互联网上大部分网络带宽被长流占用，长流的消息总字节数较大，存在突发传输。带宽测试模型中，消息字节数应该大于链路最大传输单元。

3.1.3.3 测试指标

延时 (latency) 延时指标是 DPU 完成一次请求/回复事务的传输延迟，衡量真实业务场景下 DPU 的端到端处理效率，决定最大的QPS (Query Per Second)。对于延时指标测试结果，基础网络基准测试不仅需要关注处理小数据包极限延时，还需要关注处理固定大小数据包的延迟抖动情况。

带宽 (bandwidth) 带宽指标是单位时间 DPU 传输的有效数据量，衡量数据包对链路带宽的利用能力，需要关注待测 DPU 在给定测试流量下的最大可达带宽，以及达到该带宽的饱和点，即消息大小。

单位带宽 CPU 占用率 (CPU utilization) 单位带宽 CPU 占用率是 DPU 取得 1Gbps 发送或接收带宽所消耗的宿主机 CPU 占用率，其中宿主机 CPU 占用率以流量生成器和 DPU 驱动的累积 CPU 占用率为标准。DPU 作为 CPU 的卸载引擎，使用单位带宽 CPU 占用率，可以定量衡量 DPU 卸载 CPU 网络数据包的能力。

3.1.3.4 内核态 TCP/UDP 端到端延迟测试 (N101.1basicnetbench)

输入： 消息字节数小于链路最大传输单元的短流；

输出： p50, p90 以及 p99 端到端延迟；

测试工具： Netperf；

测试描述： DPU 的基础功能是实现网络加速，TCP/UDP 是目前主要的传输层协议，通过评估内核态 TCP/UDP 端到端延迟，反映 DPU 处理请求/回复业务模型的性能。

3.1.3.5 内核态 TCP/UDP 发送带宽测试 (N102.1basicnetbench)

输入： 消息字节数大于链路最大传输单元的长流；

输出： 发送侧平均带宽以及单位带宽 CPU 占用率；

测试工具： iperf3；

测试描述： DPU 的基础功能是实现网络加速，TCP/UDP 是目前主要的传输层协议，通过评估内核态 TCP/UDP 发送带宽测试，反映 DPU 作为主机网络数据包出口的转发能力。

3.1.3.6 内核态 TCP/UDP 接收带宽测试 (N103.1basicnetbench)

输入： 消息字节数大于链路最大传输单元的长流；

输出： 接收侧平均带宽以及单位带宽 CPU 占用率；

测试工具： iperf3；

测试描述： DPU 的基础功能是实现网络加速，TCP/UDP 是目前主要的传输层协议，通过内核态 TCP/UDP 接收带宽测试，反映 DPU 作为主机网络数据包入口的承受能力。

3.2 RDMA

RDMA 是一种远程直接内存访问技术，允许用户程序绕过操作系统内核，直接和 DPU 设备交互发起数据传输，为数据中心的服务器之间提供高带宽和低处理延迟的网络通信。目前市面上的 DPU 产品普遍提供了对 RDMA 的支持，如图3.4所示。RDMA 网络可以实现数据中心内存资源的池化。上层应用通过标准化的 RDMA 动词（verbs）接口，发起 RDMA 读写操作，一台主机可以通过数据中心网络直接读写多台机器的物理内存。

3.2.1 基于 DPU 的 RDMA 加速结构

典型的RDMA 协议栈包括 InfiniBand、RoCE v1/v2 以及iWARP 等协议，RoCEv2 已经被微软的超大规模数据中心广泛部署，因此 DPU RDMA 基准测试程序以 RoCEv2 协议作为默认的待评估 RDMA 协议。图3.5给出了使用 DPU RDMA 支撑应用层数据收发的典型功能架构，DPU 直接卸载 RDMA 网络协议栈，保证了高性能。需要指出的是，如果待评估 DPU 实现了非 RoCEv2 类型的 RDMA 协议，例如亚马逊的 SRD 协议，应该保证提供标准的 RDMA 动词（verbs）应用层接口。

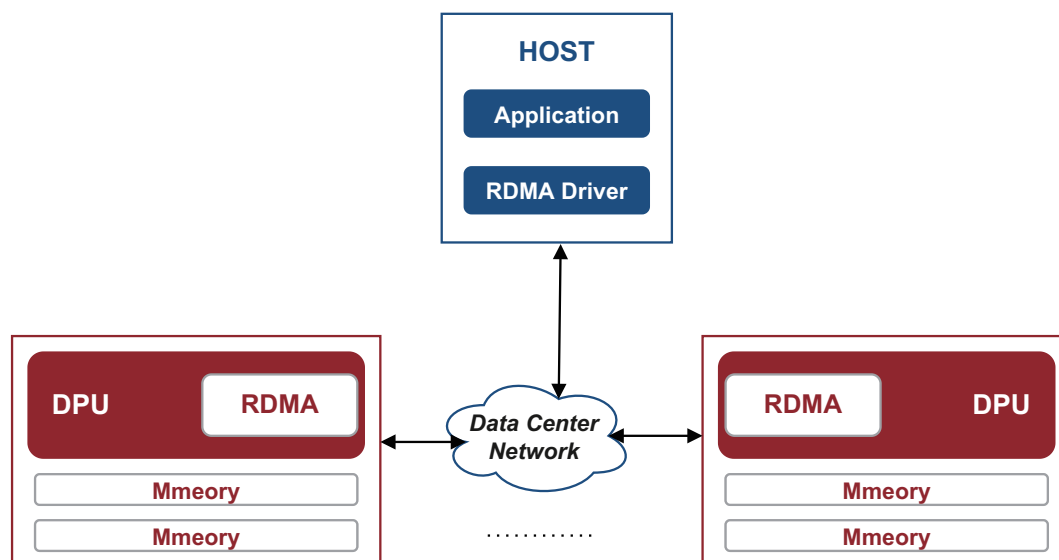


图 3.4: DPU RDMA 实现内存资源池化场景示意图

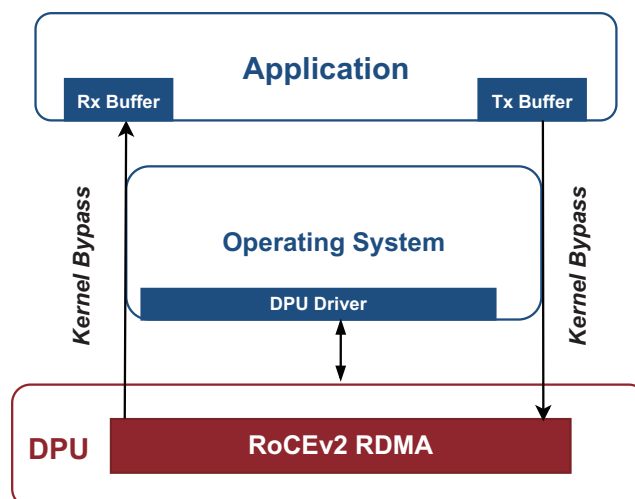


图 3.5: DPU 支撑 RDMA 应用的架构示意图

3.2.2 测试系统

测试系统拓扑采用3.1中定义的端到端测试系统，建议采用 perftest 作为 RDMA 流量生成器，Perftest 实现了 ping-pong 测试机制，提供 Send/Receive、RDMA Read、RDMA Write 以及 RDMA Atomic 操作的延迟和带宽测试，也支持多 QP 的测试。

3.2.3 Benchmark

3.2.3.1 测试负载参数

QP 类型 (QP type) 规定 QP 类型，包括 RC/XRC/UC/UD。

迭代次数 (number of iteration) 单次测试乒乓 (ping-pong) 迭代次数。

消息字节数 (message bytes) 单次乒乓 (ping-pong) 迭代过程中消息的大小，单位为 Byte。

并发 QP 数 (number of concurrent QP) Client 和 Server 之间建立的活跃 QP 连接数。

3.2.3.2 测试指标

延时 (latency) 根据待测试 RDMA 操作类型，所测延迟为应用层端到端延时 (μs) 包括以下两种定义：

- 对于 Send 和 RDMA Write 操作，延时定义为从本端 DPU 发起 Send 或 RDMA Write 操作到数据写入对端应用缓冲区的绝对时间差。
- 对于 RDMA Read 操作，延时定义为从本端 DPU 发起 RDMA Read 操作到数据写入本端应用缓冲区的绝对时间差。

相较于纯粹硬件网络协议栈层面的延迟，应用层端到端延时可以真实反映实际使用场景中的 DPU 性能。

可达带宽 (bandwidth) 可达带宽 (Gbps) 是 DPU 作为 RDMA 操作发起者或者接收方可以处理的数据速率。

单位带宽 CPU 占用率 (CPU utilization) CPU 占用率是为取得对应带宽，相关程序的 CPU 占用率。

QP 连接可扩展性 (QP extensibility) 数据中心服务器上通常运行着多种类型的服务，每种服务通常需要和多个并发的客户端通过 RDMA 进行远程的内存直接读写，因此 DPU RDMA 基准测试程序需要对 DPU RDMA 子系统可以支持的并发 QP 数量进行评

估。RDMA QP 连接可扩展性指的是在性能满足业务场景的延迟和带宽需求的情况下，待测可以支持的活跃 RDMA QP 数量。

3.2.3.3 RDMA 端到端延迟测试 (N201.1rdmabench)

输入： 发送的报文小于链路 MTU；

输出： 本次测试的最小、最大、典型、平均，p99 以及 p999 延迟；

测试工具： Perftest；

测试描述： 相对于 TCP/UDP，RDMA 能够提供最低的延迟，通过 RDMA 端到端延迟测试，可以反映待测 DPU 能否满足低延迟业务的需求。

3.2.3.4 RDMA 端到端带宽测试 (N202.1rdmabench)

输入： 发送的报文大于链路 MTU；

输出： 本次测试的平均带宽值；

测试工具： Perftest；

测试描述： RDMA 可以在零 CPU 占用的条件下，达到极高的带宽，通过 RDMA 端到端带宽测试，可以反映待测 DPU 的单位带宽 CPU 占用率。

3.2.3.5 RDMA QP 并发能力测试 (N203.1rdmabench)

输入： 发送的报文小于链路 MTU；

输出： 包率随并发 QP 数量变化的折线图；

测试工具： Perftest；

测试描述： RDMA 需要支持多个应用，每个应用可能包括多个线程，为了降低同步开销，每个线程需要一个 QP 资源，通过 RDMA QP 并发能力测试，可以反映待测 DPU 对高并发业务的支持能力。

3.3 虚拟交换机/路由器卸载加速

虚拟交换机是运行在 CPU 上通过软件方式实现的交换机，实现虚拟机之间的报文交换能力。虚拟交换机能够实现与硬件交换机相同的功能，可以有效替代硬件交换机，达到节省成本的目的。同时虚拟交换机具有配置灵活的特点，即全部功能都由软件实现，所以可以在一台服务器上虚拟出多台虚拟交换机和端口，以满足云环境中灵活多变的网络部署需求。

OVS (OpenvSwitch) 是在开源 Apache2.0 许可下的产品级多层虚拟交换机，旨在通过编程扩展，使网络实现自动配置、管理和维护，同时支持标准的管理接口和协议³。OVS 是一种支持多个物理服务器分布式的虚拟交换机。旨在利用虚拟化解决网络问题，结合控制器软件实现分布式虚拟交换技术。交换机和控制器软件能够在多个服务器之间创建集群网络配置，从而不需要在每一个虚拟机和物理主机上单独配置网络。

由于 OVS 是通过软件实现的虚拟交换设备，网络数据包的转发、存储、解析处理等均由主机端软件模拟执行，占用主机端 CPU 和内存大量资源，而这些被占用的资源并不能产生价值。而且随着网络能力要求的不断提高，OVS 要处理越来越多的网络流量，使资源消耗更加严重。DPU 可以把整个 OVS 数据平面卸载到 DPU 内部，提高了 OVS 的转发处理效率和吞吐量，降低延迟。对于用户来说 OVS 依然是虚拟交换设备，可以透明地通过 OVS 控制平面进行灵活的配置。

3.3.1 基于 DPU 的 OVS 加速结构

OVS 是软件模拟的虚拟交换机，如图3.6所示，OVS 数据平面主要包括三部分：ovsdb-server, ovs-vswitchd, ovs datapath。

ovsdb-server 是 OVS 的数据库服务器，用来存储虚拟交换机的配置信息，使用 JSON-RPC 协议与其他组件进行通信。作为 OVS 里的独立进程存在。ovs-vswitchd 是 OVS 的核心部件，实现报文上送处理以及流表查询功能，当 ovs datapath 无法处理报文的时候会吧报文上送到该进程进行完整的流表查询和转发处理，支持多个独立的网桥 (datapath)。

³如 NetFlow, sFlow, SPAN, RSPAN, CLI, LACP 802.1ag

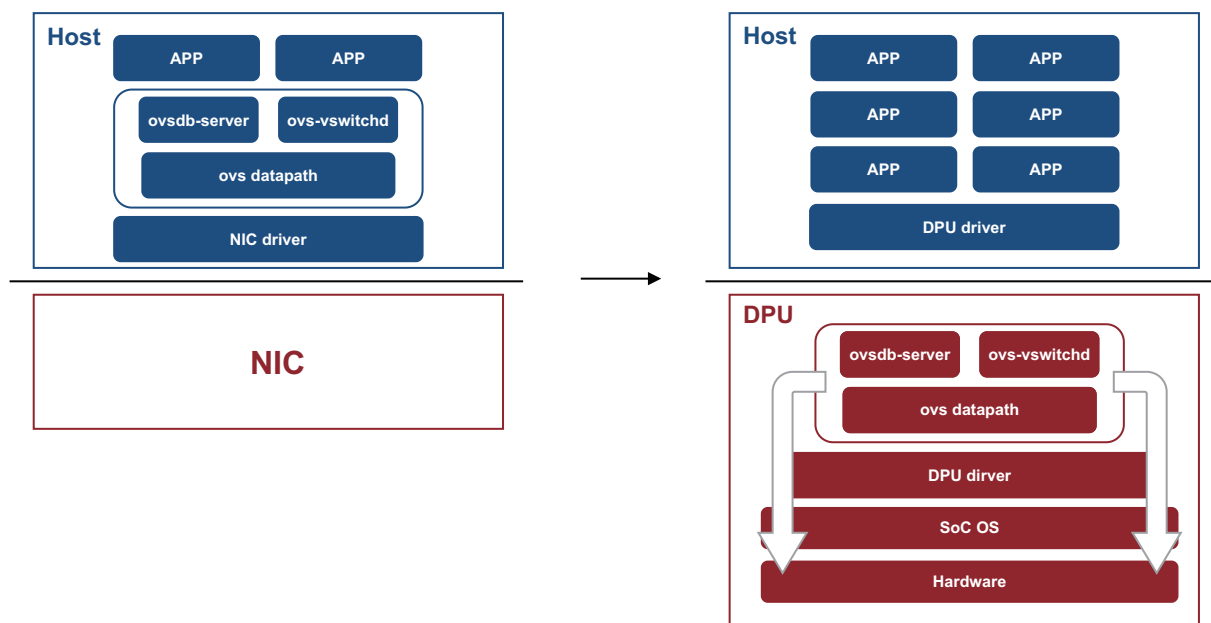


图 3.6: 基于 DPU 的 OVS 加速结构

作为 OVS 的独立进程存在。ovs datapath 是 OVS 的数据面，处理包交换和隧道，缓存数据流，如果在缓存中找到转发规则则转发，否则上送至 ovs-vswitchd 进行进一步处理。

在传统的虚拟网络架构上这三部分都要运行在主机的 CPU 上，并占用主机内存资源。在引入 DPU 后，OVS 的这三部分都可以运行在 DPU 上，能够达到节省主机资源和加速的目的。

3.3.2 测试系统

DPU OVS 基准测试系统采用端到端型测试系统，包括两个 server，其中一个 server 上部署测试工具比如 iperf3，负责发起测试报文，另一个 server 部署 DPU，并且该 DPU 具备 OVS offload 能力，该 server 还可以部署模拟 APP，用来测试 DPU 的发包能力，具体结构如图3.7所示：

3.3.3 Benchmark

3.3.3.1 测试负载参数

包大小 (packet size) 包大小是最关键的参数，影响 DPU 包处理的吞吐量和速率。包大小以 Byte 为单位。一般包大小的范围区间为 [96,9000]Bytes。理论上包越大,DPU 吞吐量越大。

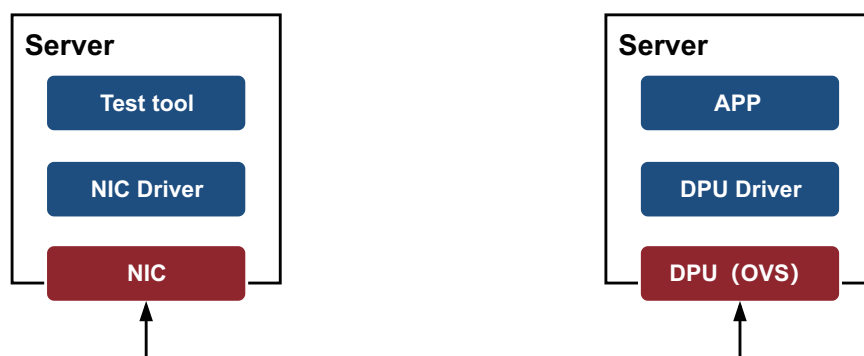


图 3.7: OVS 测试系统

收发包比例 (received/sent packet ratio) 一般来说，DPU 收发包消耗的时间不同，收包比发包消耗的时间更长。所以，对于不同应用场景有不同收发包比例。在执行测试时，不同测试用例可以有不同收发包比例。一般情况下 OVS 只是作为交换机使用，这样对于 DPU 来说收发包比例就是 1: 1。

报文封装类型 (type of packet) OVS 支持多种报文封装类型，如VLAN, VXLAN, GRE 等。对于 DPU 来说每种报文封装类型都有不同的处理时间消耗。所以 DPU OVS 功能测试例需要把报文封装类型作为基础参数。

数据流数量 (number of packet flow) 一般来说一条数据流表示五元组相同的一系列包，对于 DPU 来说同一条数据流会有相同处理步骤。因为有相同的匹配表项，所以 DPU 可以基于数据流来进行优化加速处理。但是 DPU 能够支持的数据流加速数量有限，所以在 DPU OVS 功能测试例中需要把数据流数量作为参数。

连接跟踪 (Connection Tracking, CT) 连接跟踪是跟踪并记录连接状态，OVS 本身支持连接跟踪，并且 DPU 可以实现连接跟踪功能的卸载。为了支持连接跟踪能力，DPU 需要识别五元组数据流和对应的连接，并且记录连接状态。该功能对 DPU 处理性能有很大影响，所以需要在 DPU OVS 功能测试例中把连接跟踪功能作为参数。

服务质量 (Quality of Service, QoS) QoS 能力是 DPU 的基本能力，DPU 可以支持多种 QoS 队列调度算法例如严格优先级 (Strict Priority, SP)，加权轮循 (Weighted Round Robin, WRR)，差分加权轮循 (Deficit Weighted Round Robin, DWRR) 等。不同的算法会有不同的性能消耗。所以需要在 DPU OVS 功能测试中把 QoS 队列调度算法作为参数。

3.3.3.2 测试指标

吞吐量 (throughput) 吞吐量 = 处理包的字节数/执行时间，单位是 bps。该指标把包大小也计算在内，代表 DPU 的整体包处理能力。

包速率 (Packets per second, PPS) 包速率用来计算 DPU 每秒处理的包数量。

延时 (latency) 延时代表每个包在 DPU 处理中的时间损耗。

连接建立能力 (connection per second, CPS) 连接建立能力代表 DPU 的连接建立速率，比如 TCP 创建连接并发成功率。

3.3.3.3 OVS 加速测试 (N301.1ovsbench)

输入： VXLAN 封装的报文，报文长度可变，同时适配不同被测试设备调整发包速率；

输出： 吞吐量，包速率；

测试工具： iperf3；

测试描述： 对于云网络来说 DPU 的 OVS 卸载功能能提供更好的带宽保证，而 VXLAN 隧道是云网络中的常用隧道封装，通过测试 OVS VXLAN 隧道流量，可以准确测试出 DPU 在 OVS 场景下的转发能力。

3.4 SD-WAN 网关卸载加速

软件定义广域网 (Software Defined Wide Area Network, SD-WAN) 是将软件定义网络技术应用到广域网场景中的一种服务，用于企业互联，数据中心互联及云互联等场景中，支持在混合链路 (MPLS, Internet, 5G 等) 中择优传输。如图3.8 所示，从逻辑分层及功能角度划分，SD-WAN 的逻辑架构主要包括业务呈现层、管理层、控制层以及网络层，各层之间以专用接口互通。

业务呈现层向下对接网络控制器，通过BSS (Business Support System) 和OSS (Operation Support System) 系统实现 SD-WAN 的业务呈现和发放；管理层中的网络控制器是核心组件，具有网络编排和管理功能；控制层负责网络控制；网络层主要是边缘设

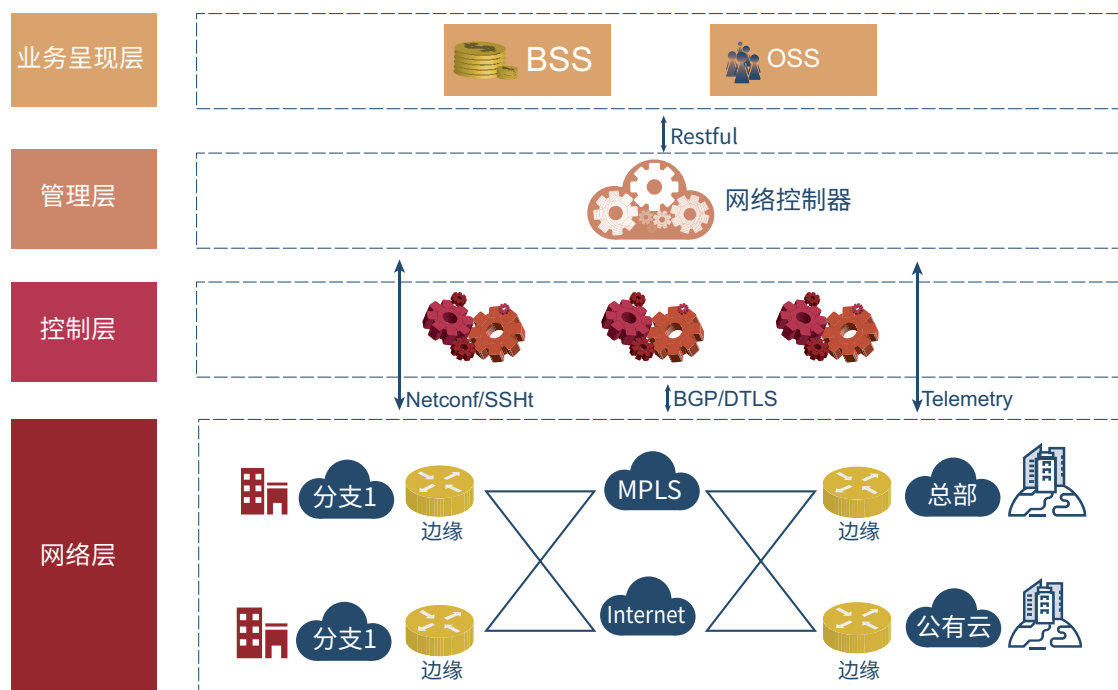


图 3.8: SD-WAN 应用场景

备 (SD-WAN Edge)，一般为出口 CPE 设备，是 SD-WAN 隧道的发起和终结点，或称为 SD-WAN 网络的边界点。

POP (Point of Presence) 是客户业务访问的接入节点，通常由 SD-WAN 供应商所提供，是由此接入到 SD-WAN 提供商的光纤主干传输网的地方。POP 节点通常采用通用服务器，并基于 NFV 及容器技术部署，运转负荷较大，POP 功能独占网卡，尤其是业务量大、性能要求高且占用资源较多的 IPsec (Internet Protocol Security) 加密隧道业务。

3.4.1 基于 DPU 的 SD-WAN 加速结构

基于 DPU 架构的 SD-WAN 加速方案，将 DPU 部署于边缘 POP 网关节点，利用 DPU 实现 IPsec 等加密算法与网络加速，从而释放服务器 CPU 资源，优化资源配置。网关 POP 节点典型业务模型如图 3.9 所示：

终端 CPE (Customer Premise Equipment) 同时建立 2 条 IPsec 隧道分别到达 POP1 和 POP2，网关之间通过 VLAN/BGP 路由以及隧道加密与对端网关互通。

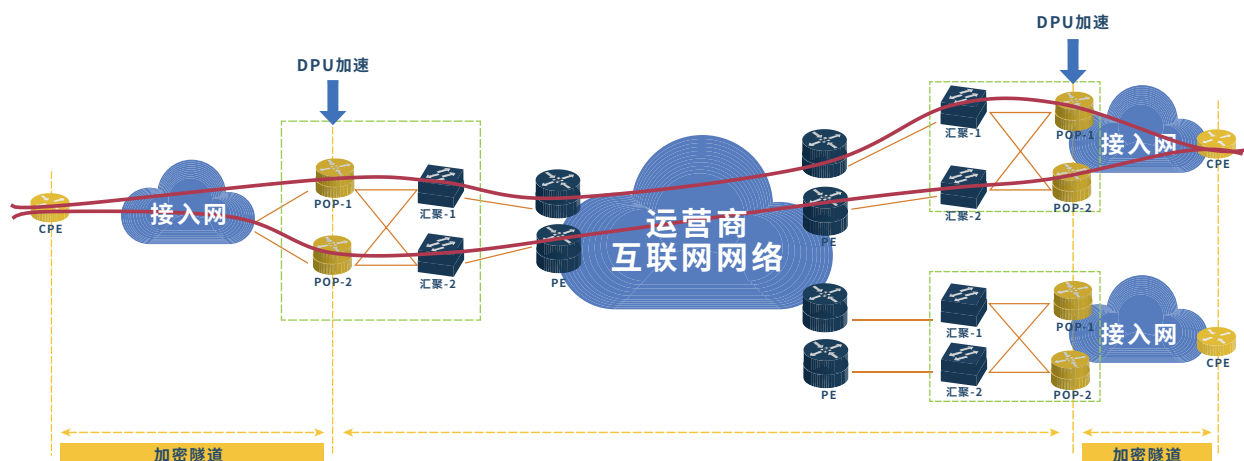


图 3.9: SD-WAN 网关节点部署

3.4.2 测试系统

DPU SD-WAN 基准测试，采用端到端型测试系统，其中，DPU 部署于网络两侧 POP 节点上，一端作为 initiator 是 IPSec 等协议的发起端，另一端系统为 target 端，作为 IPSec 的对端，两端之间建立 IPSec 隧道，实现加密通信。在 SD-WAN 网关场景下，DPU 可以按照业务需求卸载 IPSec、BGP、TCP 等网络协议转发。

搭建端到端型测试系统环境如下图 3.10 所示，主要包含设备：

1. CPE 接入终端；
2. POP 边缘节点服务器与 DPU 加速设备；
3. SDN 控制器（含软硬件）；
4. PE 等中间网络设备；
5. 测试应用程序（含软硬件）。

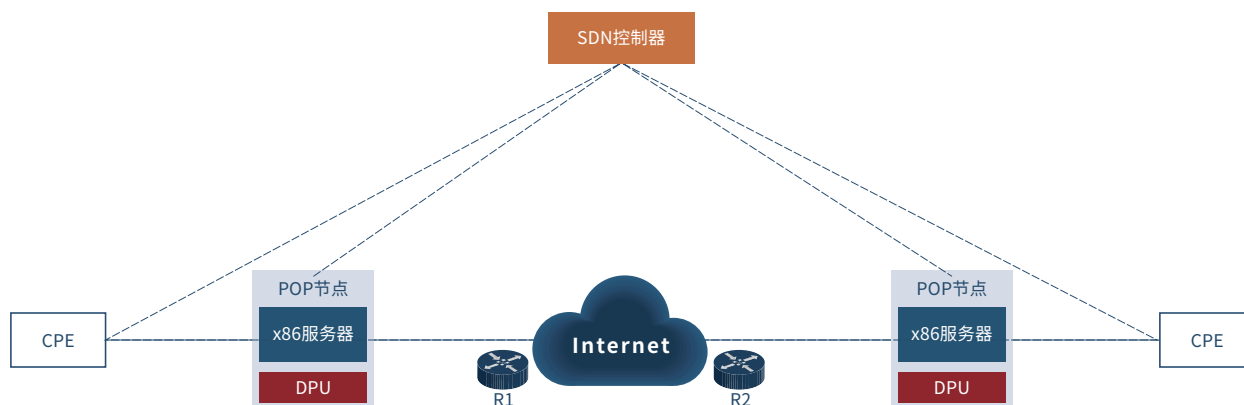


图 3.10: SD-WAN 测试系统

从组网上看，D-WAN 总体路径可以分为本端接入段（CPE 到 POP），本端汇聚段（POP 到 PE），骨干段（PE 到 PE），远端汇聚段（PE 到 POP），远端接入段（POP 到 CPE）等多段路径。一般，接入段采用 Overlay 方式的连接，骨干段为提升网络质量需要和 Underlay 协同；控制器用来计算端到端路径。

3.4.3 Benchmark

3.4.3.1 测试指标

延时 (latency) CPE 到对端 CPE 之间的建立连接后的业务时延，评估 DPU 部署后对网络时延的影响，尤其是对 IPSec 相关业务性能的影响。

抖动 (jitter) CPE 到对端 CPE 之间端到端网络抖动（最大延迟时间与最小延迟时间的差值），评估 DPU 部署后对网络稳定性的影响。

吞吐量 (throughout) CPE 到对端 CPE 之间的端到端吞吐量，评估 DPU 部署后所能满足的最大吞吐量。

丢包率 (packet loss rate) CPE 到对端 CPE 之间的端到端网络丢包率，评估 DPU 部署后对网络丢包率的影响。

会话建立时间 (session establishment time) 测试本端 CPE 到对端 CPE 之间的隧道建立的时间，用于评估 IPSec 隧道的封装和解封装性能。

3.4.3.2 SD-WAN 加速测试 (N401.1sdwanbench)

输入： 通过 SD-WAN 连接网络服务，利用测试工具产生模拟网络流量，分别测试没有部署 DPU 和部署 DPU 情况下 SD-WAN 网络性能。

输出：

1. 未部署 DPU 时，输出 TCP、UDP 网络报告及服务器 top 监测报告；
2. 部署 DPU 时，输出 TCP、UDP 网络报告及服务器 top 监测报告；
3. 对比部署 DPU 前后网络性能差异及 POP 服务器资源利用率。

测试工具： Iperf3。

测试描述： SD-WAN 为企业提供分支与分支、分支与数据中心之间的互联，提供超宽网络、按需互联、智能选路的业务体验，重塑企业 WAN 互联的业务流程，能够解决网络当前面临的 WAN 封闭架构、业务体验难保障、业务部署慢和运维困难的问题。

通过在 POP 节点上部署 DPU，测试端到端的上下行的网络质量，评估前后的网络质量差异以及服务器资源利用率。并在测试过程中，设定不同的 TCP、UDP 参数，分析测试报告，给出综合测试结论。

3.5 负载均衡 LB 卸载加速

负载均衡 (Load Balance, LB) 是一种高可用的反向代理技术，可以通过硬件或者软件的方式按照可选择的负载均衡算法把特定业务 (web 服务、网络流量等) 分配给一个或多个后端的特定服务器或设备，提高服务的并发处理能力，以及服务器等硬件资源的充分合理利用，同时也保证了服务的高可用性。负载均衡服务还能避免内部服务器 IP 地址泄露，有效的提高内部服务器的安全性。软件负载均衡运行在主机端的 CPU 上以软件实现，占用主机端 CPU 和内存资源。所以负载均衡服务软件卸载可以成为 DPU 的一种非常有价值的应用场景。利用 DPU 实现负载均衡软件卸载可以节省服务器资源同时也可以有效提高并发访问能力。

3.5.1 基于 DPU 的 LB 加速结构

基于 DPU 的负载均衡加速结构如图3.11，软件实现的负载均衡需要从普通网卡上送到主机端 CPU 处理，再通过主机端 CPU 下发到普通网卡转发到对应服务器中，而采用 DPU 实现的负载均衡，数据流不经过主机端 CPU，直接通过 DPU 转发出去或者上送主机端 CPU 进行数据处理，降低了业务相应延迟，提高了吞吐量，释放了主机端 CPU 资源。

3.5.2 测试系统

DPU LB 基准测试系统采用多端型测试系统, 包括四个 server。其中一个 server 上部署测试工具负责发起测试报文，如 perf3。另一个 server 部署 DPU，并且该 DPU 具备 LB 卸载能力。此时 LB 功能已经被卸载到 DPU，不再占用 server 端 CPU 资源，所以该

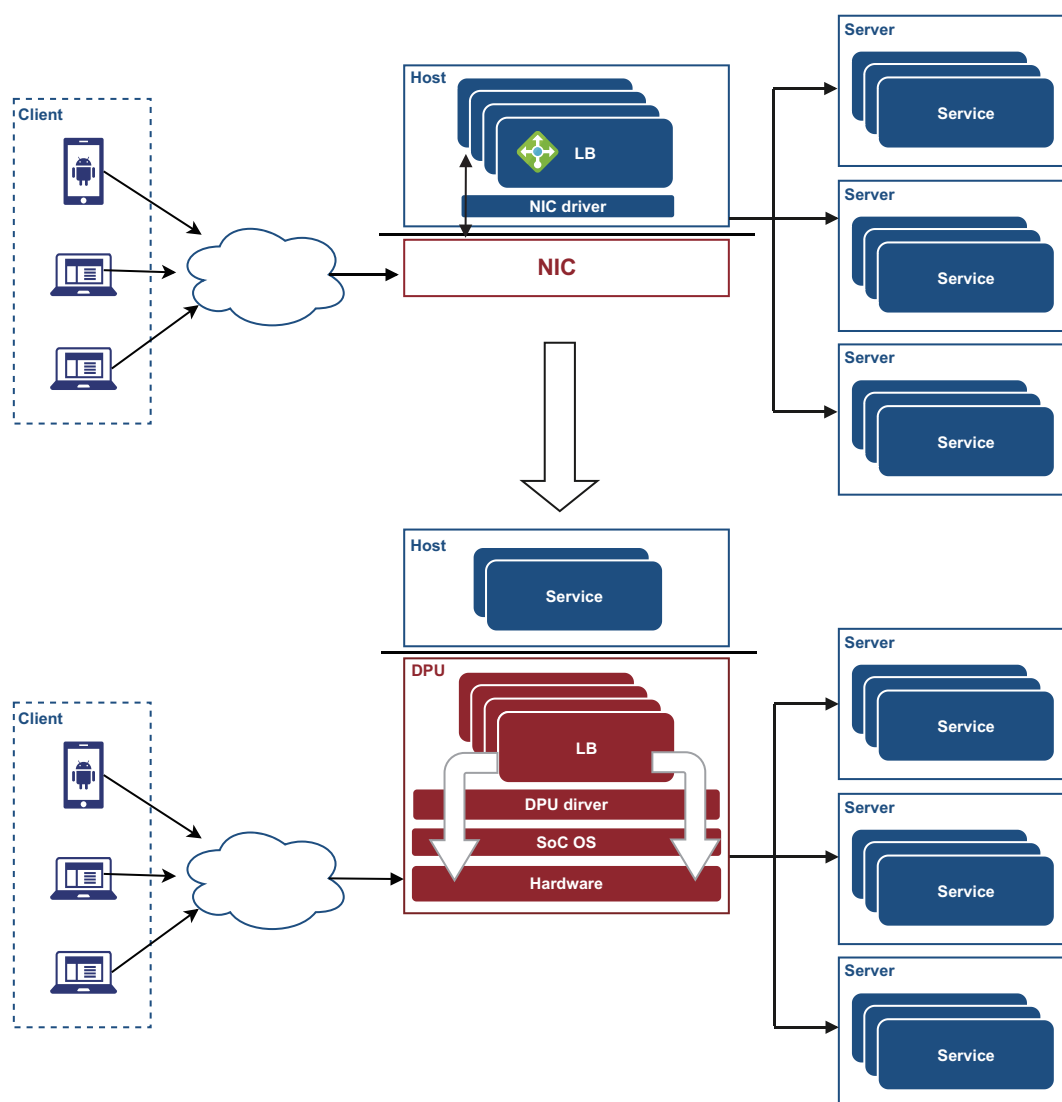


图 3.11: 基于 DPU 的 LB 加速结构

server 也可以用来部署服务 (service)。另外两个 server 部署多个服务 (service) 实例或者用测试工具模拟 server 实例，之所以使用多个 server 部署服务 (service) 实例，是为了测试在 server 能力不一致的情况下 LB 的负载均衡能力。测试系统结构如图 3.12 所示。

3.5.3 Benchmark

3.5.3.1 测试负载参数

负载均衡类型 (layer of load balance) 负载均衡分为二层、三层、四层、七层这几类。其中四层负载均衡和七层负载均衡是最常用的负载均衡类型。四层负载均衡工作在 OSI

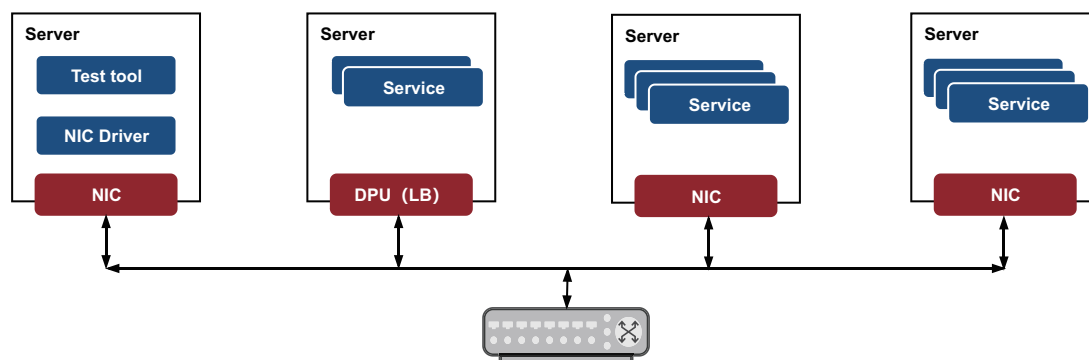


图 3.12: LB 测试系统

模型的传输层，针对 TCP/UDP 报文进行负载均衡。七层负载均衡工作在 OSI 模型的应用层，应用层协议较多，常用 HTTP、RADIUS、DNS 等。由于四层和七层负载均衡实现差异很大，所以 DPU LB 基准测试需要把负载均衡类型最为最重要的参数。

负载均衡工作方式 (type of load balance) 负载均衡服务有两种工作方式：网络地址转换 (Network Address Translation, NAT) 和直接路由 (Direct Routing, 简称DR)。网络地址转换方式由于需要进行网络地址转换，对负载均衡服务的性能损耗更大，反之直接路由方式则会有效减少负载均衡服务的负载，有效的避免负载均衡服务成为网络瓶颈。负载均衡工作方式也要作为 DPU LB 基准测试的重要参数。

负载均衡算法 (load balance algorithm) 负载均衡服务器在决定把外部请求发送到哪台 server 或哪个服务 (service) 实例时，通过负载均衡算法确定。负载均衡算法包括：轮询方式 (Round Robin)，比例 (Ratio)，优先权 (Priority)，最少的连接方式 (Least Connection)，最快模式 (Fastest) 等。由于每种算法的实现方式不同，影响负载均衡效果，所以 DPU LB 基准测试需要把负载均衡算法作为重要参数。

并发连接个数 (number of concurrent connection) 该参数代表负载均衡服务承载了多少条连接，该个数是负载均衡服务最重要的测试参数。

负载均衡服务实例个数 (number of service) 服务 (service) 实例个数代表负责均衡服务需要像多少个服务实例分发服务请求，该参数会影响服务均衡的效果。

3.5.3.2 测试指标

并发最大连接数 (maximum number of concurrent connections) 并发最大连接数是指在四层或者七层负载时，一个负载均衡实例能够承载的最大连接数量。

连接建立能力 (Connection per Second, CPS) CPS 是指在四层或者七层负载均衡时，新建连接的速率。

每秒请求数 (Query per Second, QPS) QPS 是指仅在七层负载均衡时，每秒处理 HTTP/HTTPS 的查询数量。

带宽 (bandwidth) 带宽 (Mb/s) 是单位时间内处理报文总数据量，代表负载均衡实例处理报文的实际性能。

延时 (latency) 延时代表每个报文在 DPU 处理中的时间损耗。

3.5.3.3 LB 加速测试 (N501.1lbbench)

输入： 根据被测试设备能力建立多条 TCP 连接，并为每条连接触发均匀的 TCP 流量。

输出： CPS，带宽。

测试工具： Iperf3。

测试描述： LB 主要是分为四层负载均衡和七层负载均衡，两种负载均衡类型的基本测试指标都是所以对连接建立能力 (CPS)，用来衡量负载均衡设备对多条连接进行均衡的负载分配能力，所以对连接建立能力 (CPS) 的测试可以作为 DPU LB 卸载功能的主要测试例。

3.6 5G MEC

边缘计算是云计算一种补充服务，将应用服务从集中式数据中心移动到网络边缘节点。其应用程序在边缘侧发起，使得边缘计算能够满足 5G 的关键性能指标，尤其是在低时延和高带宽方面。5G MEC (Multi-access Edge Computing, MEC) 是一种边缘计

算部署架构，MEC 平台将计算能力、业务以及部分5G UPF（User Plane Function，UPF）网络能力部署到网络边缘，实现大流量、低时延、敏感数据等本地处理，主要应用于自动驾驶、工业控制、AR/VR 等业务场景。

在 5G MEC 中下，DPU 可以在 OVS（OpenvSwitch）和 5G UPF 两个方面提升网络能力（如图3.13所示），从而为既有系统提供更加卓越的性能和体验质量。

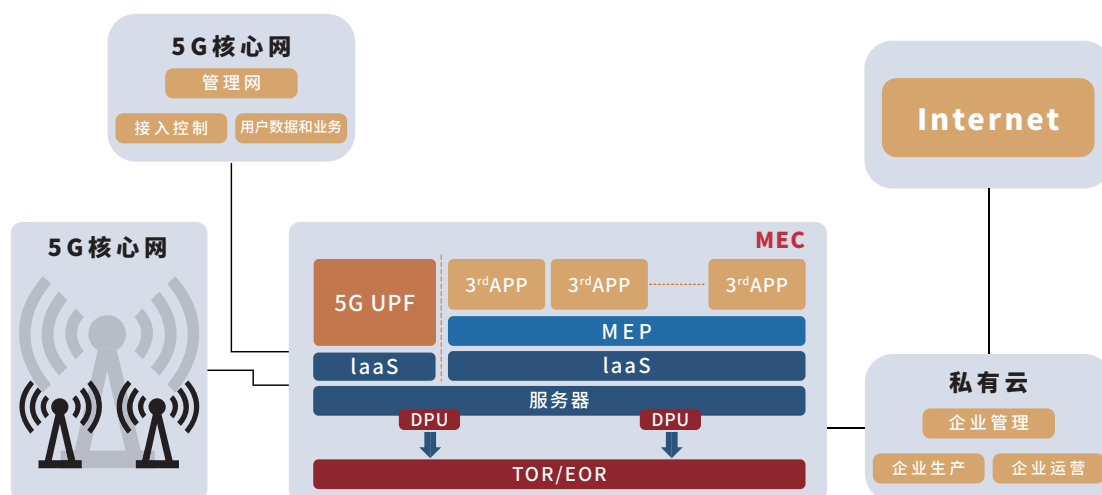


图 3.13: 5G MEC 中的 DPU

1) OVS（OpenvSwitch）加速：MEC 应用对基础设施的低延时、高速率转发性能有很高的要求，在 NFVI 层引入 DPU 可突破转发性能和数据处理能力瓶颈。ovs-vswitch 控制面接受 SDN 控制器的流表配置消息，将流表配置写入数据库，更新到 OVS 数据面。OVS 数据通道下沉到 DPU，将转发卸载到 DPU，释放 CPU 资源。

2) 5G UPF 加速：用户面 UPF 以虚拟网元 NFV（Network Functions Virtualization，NFV）的形式下沉到 MEC，满足降低传输时延，减少带宽占用，实现软硬解耦的需求。但由于通用服务器保证通用性，而丧失了专用性，即不擅长处理特定任务，比如处理编解码转换、报文转发、加解密等并行处理任务，从而相对于传统专用硬件性能有所下降，对于超低时延的应用和业务无法满足其性能要求，引入 DPU 作为 UPF 硬件加速功能，与 5G 技术相结合，可以为边缘计算创建强大的运行环境。

3.6.1 基于 DPU 的 5G MEC 加速结构

5G MEC 包括 5G 网关 UPF、边缘应用平台 MEP、DPU 加速硬件、行业应用 APP 以及虚拟化基础设施。总体架构⁴如图 3.14。

⁴<https://www.etsi.org/>

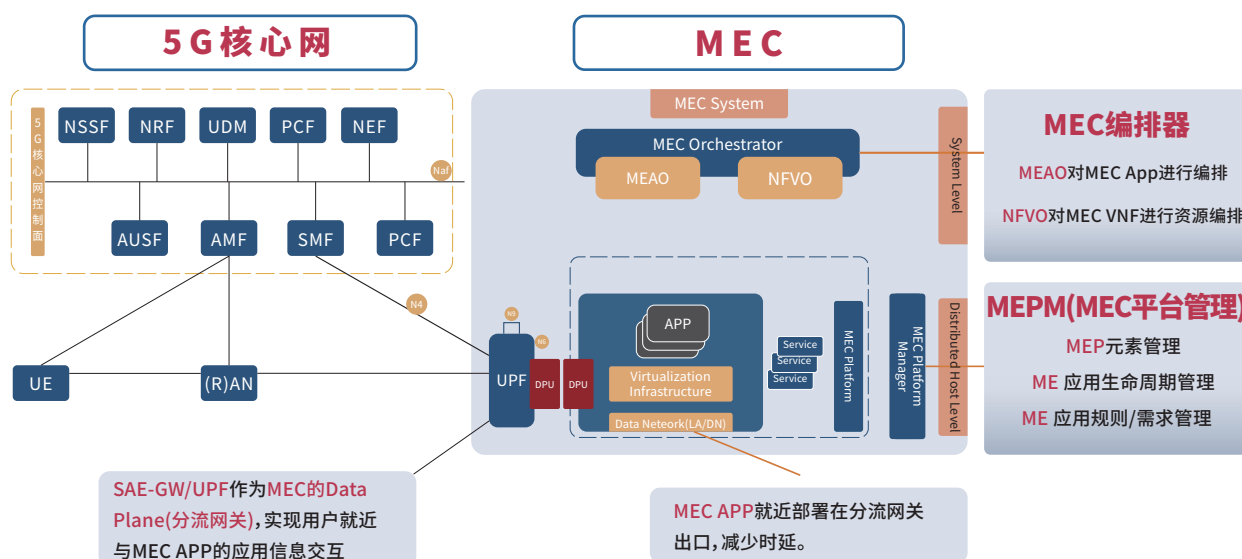


图 3.14: 5G MEC DPU 加速结构图

5G MEC 部署在靠近终端用户侧的边缘位置，提供大带宽、低时延的网络连接能力，AI、图像渲染等计算能力，以及面向行业的安全能力。在运营商实际网络中，MEC 可以基站、传输节点，本地 UPF 等多种网元共部署。尽管 MEC 可以灵活地部署在从基站附近到数据中心网络的不同位置，但是所有部署场景都基于 UPF，利用 UPF 引导流量流向目标 MEC 应用程序和网络。

3.6.2 测试系统

DPU MEC 基准测试系统，采用端到端型测试系统，其中，DPU 部署于边缘计算网络两侧对应网元节点上，其中一端位于 UPF 服务器节点作为发起者（initiator）负责网关侧加速端，另一端部署于 MEC 服务器（target），负责云计算侧加速，二者共同作用实现网络的上下行转发加速，提升端到端数据处理性能。

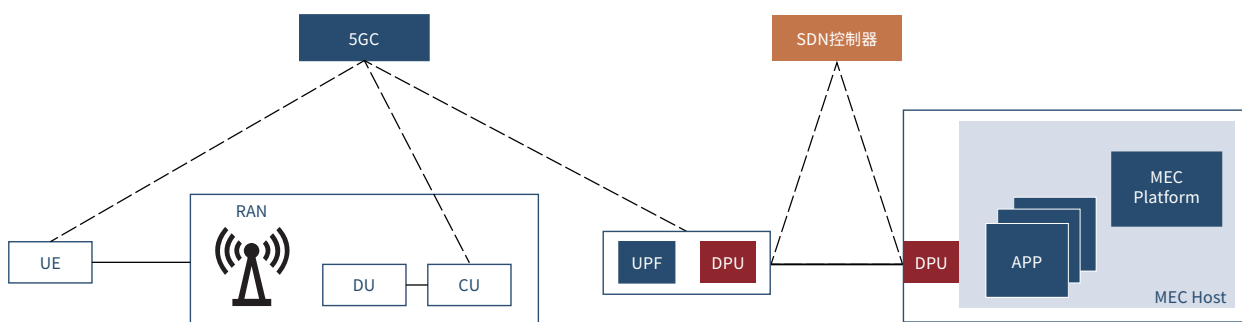


图 3.15: 5G MEC DPU 测试组网

搭建端到端型测试系统如图3.15，主要包含设备：

1. 接入终端用户设备端（User EquipmentUE）；
2. 5G 无线接入网 5G RAN（Radio Access Network）；
3. 5G 核心网 5GC；
4. SDN 控制器（含软硬件）；
5. UPF 与 DPU 加速设备；
6. MEC 与 DPU 加速设备；
7. 测试应用程序。

3.6.3 Benchmark

3.6.3.1 测试指标

延时 (latency) 用户设备端到 MEC APP 之间的延时，用于评估 DPU 部署后对网络时延的影响。

抖动 (jitter) 用户设备端到 MEC APP 之间的抖动（最大延迟与最小延迟之间的差值），用于评估 DPU 部署后对网络稳定性的影响。

吞吐量 (throughout) 用户设备端到 MEC APP 之间单位时间内传输的数据量，用于评估 DPU 部署后最大吞吐量。

会话建立时间 (session establishment time) 测试无线接入网络与数据网络 (DN, MEC) 之间会话建立的时间，用于评估用户面GTP 隧道协议的封装和解封装性能。

3.6.3.2 5G MEC 加速测试 (N601.15gmecbench)

输入： 通过 5G 网络终端，访问 MEC 平台部署的相关业务（用测试工具生成模拟网络流量），测试网络性能；分别进行没有部署 DPU 和部署 DPU 的场景测试。

输出：

1. 未部署 DPU 时，输出 TCP、UDP 网络报告；
2. 部署 DPU 时，输出 TCP、UDP 网络报告；
3. 对比部署 DPU 前后网络性能差异。

测试工具： Iperf3。

测试描述： 通过在 MEC 和终端上部署相关应用 (视频、FTP 类等) 及测试工具，测试上下行的网络质量，评估部署 DPU 前后的网络质量差异。并在实际测试过程中，设定不同的 TCP、UDP 参数，分析测试报告，给出综合测试结论。

第 4 章 面向存储的基准评测

存储是数据中心提供的主要服务之一。为提高存储资源的利用率，数据中心利用网络将存储系统通过网络互连并直接管理，从而实现存储资源的按需分配。在这样的结构中，存储性能影响因素主要有三方面：主机处理存储网络协议栈性能、互连网络性能和存储介质性能。随着数据中心的存储介质从机械硬盘逐渐向固态硬盘转变，尤其是支持 NVMe 协议的固态硬盘的采用，以至接近于内存访问延时的存储类内存（SCM）存储介质的访问延时从毫秒量级缩短到几十甚至几微秒；高速无损互连网络（RDMA）在数据中心也得到广泛部署，互连网络性能得到极大提升，这使得存储性能瓶颈从存储介质、网络传输逐渐转移到主机侧处理存储网络协议栈的处理。主机侧虽然可以通过运行 SPDK 这样的用户态程序，通过轮询减少中断，降低上下文切换的开销，提高存储性能，但轮询使得主机侧 CPU 无法服务于其他业务，造成额外的开销，并不是理想的解决方案。

DPU 能够有效解决存储性能遇到的瓶颈。首先，DPU 将网络与存储协议从主机侧 CPU 中剥离，降低对主机侧 CPU 的占用率、提高处理效率；DPU 集成的专用计算加速单元，如存储中计算密集的加解密与压缩解压缩单元，能够加快对存储数据的处理。其次，DPU 通过 SR-IOV 等硬件虚拟化方式支持存储虚拟化，将单个设备虚拟化成多个设备，这样虚拟机可以直接与硬件存储设备交互，减少数据拷贝和搬运次数、降低访问延迟，从而提高虚拟化环境下存储访问的性能。() DPU Benchmark 存储性能的评测选择 NVMe-oF 作为存储的基准测试，用来表征 DPU 对于存储性能的加速能力。数据加密、压缩、检错纠错等和存储也相关的内容将作为未来的工作在后续的版本中纳入。

4.1 NVMe-oF 加速

NVMe 协议是一种先进的主机与存储介质接口协议，具有多队列、低时延、高吞吐等特点。基于 NVMe 原生提出的 NVMe-oF（NVMe over Fabric）可以使 NVMe 从支持本地存储（DAS）发展为支持网络存储（NAS）且无需转换其他存储协议，在网络存储中延续保持 NVMe 存储访问低时延、高吞吐的特点。

4.1.1 基于 DPU 的加速结构

传统的存算分离结构如图4.1左所示，主机侧 CPU 至少需要运行三层协议栈（如果不是相同的存储协议，需要增加第四层协议栈进行协议转换，以 NVMe 为例，需要将 NVMe 命令转换为 iSCSI 命令才能进入网络传输层），存储协议栈、存储网络层、传输层协议栈，最终通过网卡转发出去。在基于 DPU 的存储架构中，如图4.1右所示，主机侧只负责发出存储命令，即只需要运行一层存储协议栈。其他协议栈将卸载到 DPU 中执行，降低主机端 CPU 的占用率。同时通过 DPU 具有的专用加速单元，加速存储数据的处理。

在基于存算分离的架构中，NVMe-oF 是主要的远程存储访问协议，现阶段的 DPU 产品都支持 NVMe 及相关协议的卸载。根据实现方式不同，NVMe-oF 的加速方案可分为分为半卸载与全卸载两种。

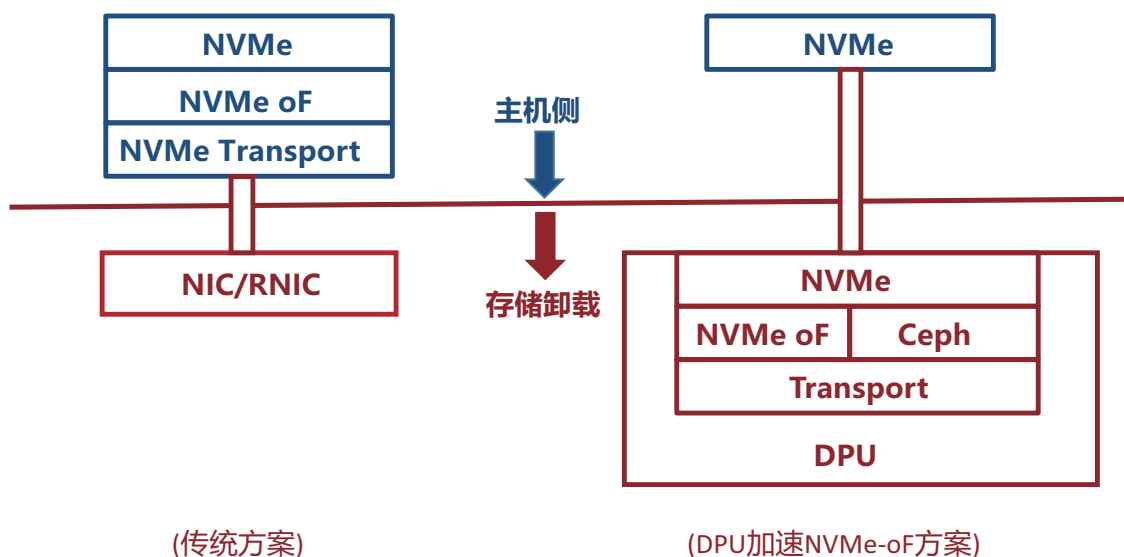


图 4.1: 加速结构

4.1.1.1 半卸载 (Partial Offload)

DPU 存储半卸载指将原运行在主机端的存储协议栈卸载到 DPU 中的 CPU 核心中处理，结合 DPU 的专用加速单元如加解密、压缩解压缩实现存储的加速。DPU 对主机而言是一块或者多块 NVMe SSD，主机直接将存储命令发送给 DPU，DPU 将存储命令转移到 DPU 中的 CPU 核心中处理，解析 NVMe 存储命令后，可以转换成不同的存储后端（分布式存储、NVMe SSD 存储系统、厂商自定义存储系统等）命令，最终经过网

络处理器（Network Processor, NP）发送出去；或者接收远端存储系统发来的读写命令，将数据搬运到主机内存中。

在 DPU 存储半卸载中，主机侧软件不需要管理网络协议（TCP、RDMA、FC），也不需要转换存储命令到相应的网络层。主机侧可以直接发送存储命令，降低网络存储对主机侧 CPU 的占用率。同时，DPU 通过加解密、压缩解压缩等专用单元可以提高数据访问时的计算速度，利用 DPU 中通用的 CPU 核心可兼容多种存储后端系统，如图4.2所示。

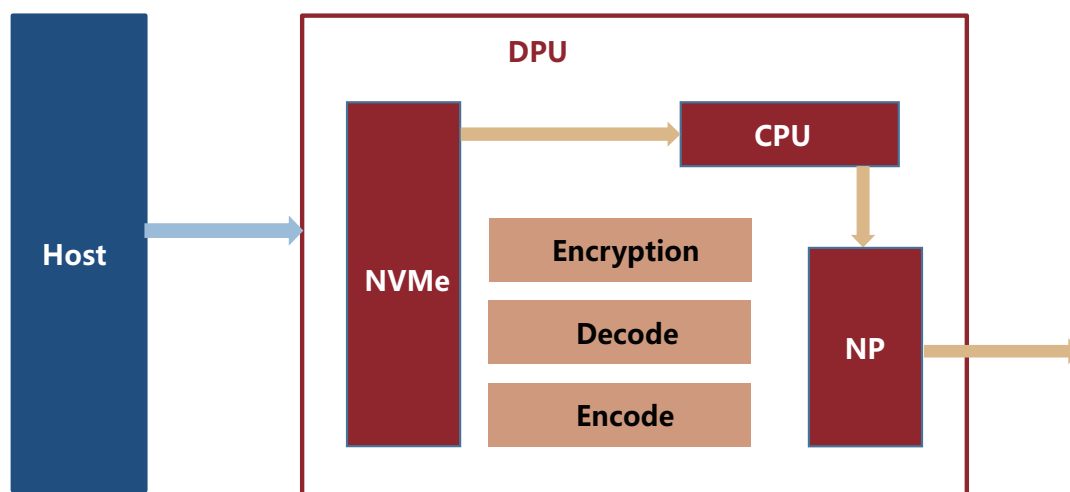


图 4.2: NVMe-oF 半卸载示意图

4.1.1.2 全卸载 (Full Offload)

DPU 存储全卸载仍然将运行在主机端的存储协议栈转移到 DPU 中执行，但 DPU 中的 CPU 核心只负责配置存储控制器的参数，例如，IO 队列数、队列深度、可并发命令数等。在 DPU 存储全卸载的模式下，主机发起的存储命令将直接通过 DPU，经由网络卸载引擎直接转发出去。类似的，接收网络传来的数据直接经过后端 DPU 的存储加速单元写入主机内存。

在 DPU 全卸载的结构如图4.3所示，DPU 的 CPU 核心只需要执行管理命令；存储命令的转换、数据的转发与写入、加解密、压缩解压缩都通过专用的存储加速单元完成，进一步降低存储访问延时同时提高存储访问的并行度。

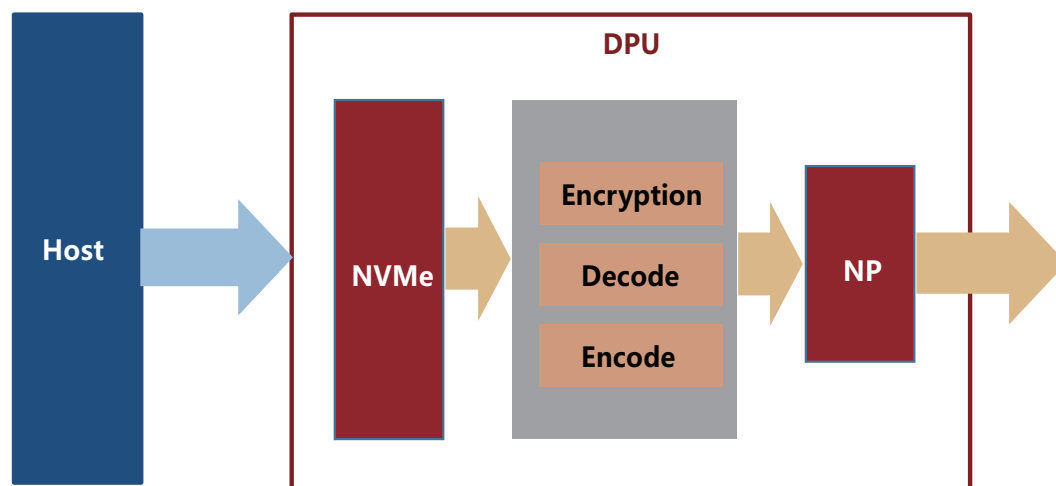


图 4.3: NVMe-oF 全卸载示意图

4.1.2 测试系统

如上所述，DPU 负责处理 NVMe-oF 协议及网络传输层协议，对用户表现为本地 NVMe SSD，即网络协议对用户是透明的，所以 NVMe oF 基准测试从用户的角度出发，关注主机端 DPU 模拟 NVMe SSD 的性能。DPU NVMe oF 基准测试采用端到端型测试系统，结构如图4.4所示。其中，一个 DPU 系统作为存储命令的 initiator，另一个 DPU 系统作为存储的 target。

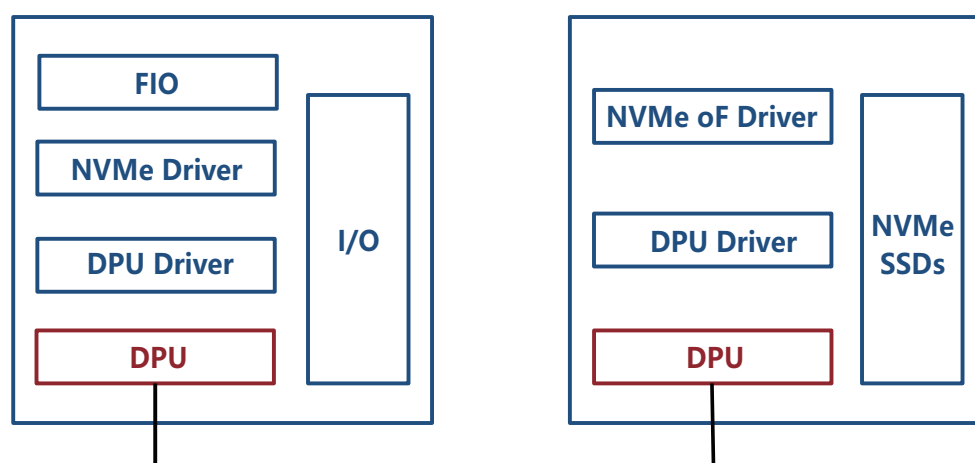


图 4.4: 测试系统

作为存储命令发起端的 DPU 系统，需要满足以下条件：

1. 主机系统需要具有较多的 CPU 核心数（128、256，测试并行的存储请求）；
2. 安装 NVMe-oF 基准测试相关的负载生成器（FIO）；

3. 操作系统需要满足运行 NVMe 驱动、DPU 驱动。

作为存储的后端的 DPU 系统，需要满足以下条件：

1. 需要通过 PCIe 总线接入多组 NVMe SSD（大于等于 DPU 支持的 NVMe 存储盘个数）；
2. 操作系统需要满足运行 NVMe 驱动、DPU 驱动。

4.1.3 Benchmark

4.1.3.1 测试负载参数

数据量（transfer size） 数据大小是一条存储命令包含的数据量，单位为 KB。

读比例（read fraction） 在一个指定负载中，存储命令中读命令所占比例。

数据模式（data pattern） 数据模式分为两种，随机访问与顺序访问：在顺序访问中，存储命令的数据地址顺序增加或减少；随机访问中，存储命令的数据地址无规律。数据 I/O 的模式灵活，每一种应用都有其特殊的数据访问模式。基于代表性与可测性的约束，只考虑随机访问与顺序访问。

并行命令数（outstanding I/O requests） 并行命令数即最大的并发命令数量。在测试场景中等同于 NVMe IO 队列的深度。

队列数（number of queue） 队列数即主机端使能 NVMe IO 队列的数量。

4.1.3.2 测试负载模型

根据存储命令数据量（大数据、小数据）与数据模式（随机、顺序）定义四种测试负载模型：第一类为大数据、顺序的数据搬移，例如大型文件处理、数据库查询；第二类为大数据、随机的数据搬移，例如 web 查询、key-value 查询；第三类为小数据、顺序的数据搬移；第四类为小数据、随机的数据搬移。第三类与第四类测试负载主要测试裸盘性能，合并为一种测试模型。每种测试负载模型中，使用 FIO 或者 IOZone 工具生成。

大数据顺序读写负载模型 大数据、顺序的数据搬移选取大文件处理、数据库查询作为典型应用，模拟其存储命令规律，参考的测试参数如表4.1所示。

表 4.1: 大数据顺序读写负载模型

测试名称	数据大小 (KB)	读比例	数据模式	并行命令数
大文件处理 1	1024	0.0	顺序	1
大文件处理 2	256	0.0	顺序	1
大文件处理 3	1024	0.5	顺序	1
大文件处理 4	256	0.5	顺序	1
大文件处理 5	1024	1.0	顺序	1
大文件处理 6	256	1.0	顺序	1
数据库查询 1	1024	0.99	顺序	4
数据库查询 2	1024	0.99	顺序	1
数据库查询 3	64	0.99	顺序	4
数据库查询 4	64	0.99	顺序	1

大数据随机读写负载模型 大数据、随机的数据搬移选取 web 搜索、key-value 查询作为典型应用，模拟其存储命令规律，参考的测试参数如表4.2所示。

表 4.2: 大数据顺序读写负载模型

测试名称	数据大小 (KB)	读比例	数据模式	并行命令数
web 搜索 1	512	0.0	随机	64
web 搜索 2	128	0.0	随机	128
web 搜索 3	512	0.5	随机	64
web 搜索 4	128	0.5	随机	128
web 搜索 5	512	1.0	随机	64
web 搜索 6	128	1.0	随机	128
key-value 查询 1	1024	0.5	随机	64
key-value 查询 2	1024	0.5	随机	128
key-value 查询 3	64	0.5	随机	64
key-value 查询 4	64	0.5	随机	128

小数据读写负载模型 小数据、顺序的数据搬移与小数据、随机的数据搬移选取测试裸盘性能常用的存储负载模式。参考的测试参数如表4.3所示。

4.1.3.3 测试指标

每秒读写次数(IOPS) 该指标为单位时间内,DPU 在给定负载下执行的读写次数, $IOPS = \text{特定负载读写次数} / \text{执行总时间}$ 。

吞吐量 (throughput) 该指标为单位时间内, DPU 在给定负载下读取与写入的数据量总和, $\text{吞吐量} = (\text{读数据量} + \text{写数据量}) / \text{执行总时间}$ 。

表 4.3: 小数据读写负载模型

测试名称	数据大小 (KB)	读比例	数据模式	并行命令数
裸盘数据读写 1	0.5	0.3	顺序	1
裸盘数据读写 2	0.5	0.3	随机	1
裸盘数据读写 3	4	0.5	顺序	1
裸盘数据读写 4	4	0.5	随机	1
裸盘数据读写 5	32	0.9	顺序	1
裸盘数据读写 6	32	0.9	随机	1

延时 (latency) 延时分为读延时与写延时，读延时为 DPU 将数据从存储介质搬运到应用侧的时间，写延时为 DPU 将应用侧产生的数据写入存储介质的时间。

4.1.3.4 裸盘性能测试 (M101.1nvmeofbench)

输入： 负载模型一、负载模型三，根据不同的负载模型，可以使用 FIO 生成指定的命令格式；根据 DPU 支持的最大队列数（假设为 n ），分别测试队列数为 1， $n/4$ 、 $n/2$ 、 $3n/4$ 、 n 时 DPU 的 NVMe 性能；

输出： 每一个模型中的利用 FIO 生成测试都将输出最大、最小、平均延迟，取平均延迟；取 FIO 输出 IOPS 与吞吐量；

测试工具： FIO；

测试描述： DPU 通过模拟 NVMe SSD（DPU 负责将 NVMe 命令转换为命令，并通过配置不同的网络传输层（TCP、RDMA）发送或接收）对主机提供网络存储服务。通过 FIO 测试工具可以生成相应的负载模型，对搭建好的 DPU 测试系统进行测试。裸盘性能测试不经过文件系统，直接进行存储命令级别（NVMe IO 命令）的测试，测试 DPU 的存储性能。

4.1.3.5 虚拟化存储性能测试 (M102.1nvmeofbench)

输入： 负载模型一、负载模型二、负载模型三，根据不同的负载模型，可以使用 FIO（在虚拟机中）生成指定的命令格式，每个模型中的每个测试分别执行；根据 DPU 支持的虚拟 NVMe 最大队列数（假设为 m ），分别测试队列数为 1， $m/4$ 、 $m/2$ 、 $3m/4$ 、 m 时 DPU 的 NVMe 性能；

输出： 每一个模型中的利用 FIO 生成测试都将输出最大、最小、平均延迟，取平均延迟；取 FIO 输出 IOPS 与吞吐量；同时检测主机端 CPU 的占用率；

测试工具： FIO；

测试描述： DPU 存储虚拟化中可以采用 SR-IOV 技术。在 SR-IOV 技术中，PCIe 设备分为 PF(physical function) 与 VF(Virtual function)，PF 是真实的存储设备，VF 是虚拟存储设备。在进行虚拟化测试时，测试过程增加了配置虚拟 NVMe 设备、创建虚拟机的步骤。假设 DPU 支持的虚拟存储设备数为 p ，分别测试开启 $p/4$ 、 $p/2$ 、 $3p/4$ 、 p 时的存储能力。虚拟化存储性能测试主要测试在 DPU 在 NVMe 虚拟化开启时，虚拟机访问磁盘性能。

第 5 章 面向计算的基准评测

本书所指的计算主要是直接执行应用层的计算能力，或者为应用提供计算平台或统一计算资源的数据处理能力。算力可分为通用算力和专用算力。通用算力主要涉及 PaaS 层的负载，比如 Spark，服务网格等，对于底层的网络性能依赖比较高；而专用算力主要是 SaaS 层的负载，面向计算密集的算法加速场景，通过 DPU 内的专用硬件、专用处理单元提供专用算力。例如，在 Marvell 的 DPU 产品中，集成的 AI 推理单元就是一种专用算力。在早期的智能网卡（SmartNIC）设计中，一定程度上实现了对数据密集型应用提供专用算力支持，包括数据库中的查询加速，AI 训练推理，视频/图像编解码等。在这些计算密集的数据处理系统中，依靠通用 CPU 计算存在严重的性能瓶颈。作为专用算力的提供设备，DPU 支持网络与存储加速功能，在进行数据处理时更接近数据源（直接处理从网络中或者存储中获得数据，不需要经过主机的干预，缩短数据传输路径），相比于 GPU 及其它专用算力提供设备（如加速不同专用计算的 XPU，基于 FPGA 的可编程加速方案等），具备更强的数据处理能力。DPU Benchmark 计算部分分别选取常见数据密集型计算中的典型应用为基准测试，包括：分布式计算系统中 Spark，AI 计算等。尽管本章中涉及的测试场景尚未有单一的 DPU 产品完全支持，但对未来 DPU 设计仍然具有一定参考意义。

5.1 Spark 加速

Spark 是一种用于大数据处理的分布式计算引擎，基于 MapReduce 编程模型及分布式计算架构实现了海量数据的高性能并行计算。通用计算处理器 CPU 提供 Spark 系统的基础算力，受限于 CPU 的计算性能，基于 CPU 的 Spark 的计算性能出现瓶颈。在数据处理和计算过程中，数据在系统内存和磁盘之间多次搬运，产生大量磁盘 I/O，低效的磁盘访存也是限制 Spark 性能的瓶颈之一。此外，Spark 作为分布式系统，节点之间的数据搬运效率依赖网络 I/O，网络性能决定了 Spark 数据传输的性能。作为一种典型的大数据量计算密集型应用，Spark 的性能由于 CPU 算力、磁盘 I/O 及网络 I/O 的限制而遇到增长瓶颈，难以满足日益增长的计算需求。

通过 DPU 卸载计算，可以一定程度上解决上述三个方面的性能瓶颈，从而实现对 Spark 的加速，包括 Spark 算子的 DPU 卸载、磁盘 I/O 的 DPU 卸载及分布式环境下的

网络 I/O 卸载。DPU Spark 基准测试主要评测在真实 Spark 环境中 DPU 对 Spark 计算的加速效果，即综合评测 DPU 在计算、存储 I/O 和网络 I/O 时算力卸载的性能。

5.1.1 基于 DPU 的 Spark 加速结构

基于 DPU 的 Spark 加速结构如图5.1所示，通过在 Spark 集群中的工作节点安装 DPU 设备，将 Spark 任务执行过程中最耗时的复杂算子计算、本地存储读写及跨节点的数据同步卸载到 DPU 中，从而加速 Spark 任务。

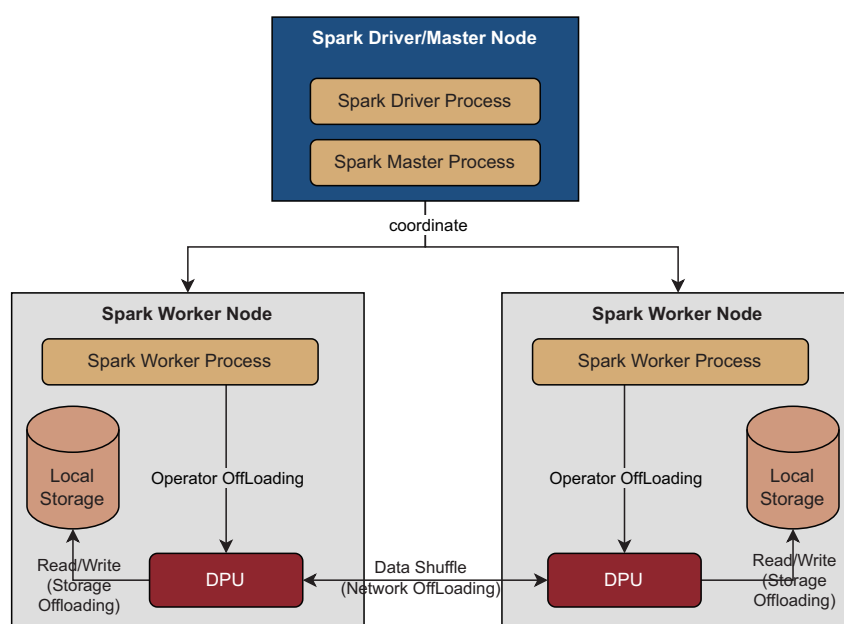


图 5.1: 基于 DPU 的 Spark 加速系统架构图

5.1.2 测试系统

Spark 作为典型的分布式系统，其测试系统为多端型测试系统，DPU 的数量等于 Spark 集群中工作节点的数量。

图5.2给出一种多端型测试系统的连接实例。选择 A、B 两套具有相同基本物理配置的 Spark 集群，A 为无 DPU 加速的 Spark 集群系统，B 为 DPU 加速的 Spark 集群系统。系统的输入为测试数据集以及基准测试程序；输出为 Spark 测试任务的总执行时间。通过选择不同测试数据集、不同 Spark 测试任务以及与之匹配的集群规模完成测试。Spark 应用场景的数据量大，要求测试数据集的规模应该反映实际数据量，以常用的分布式文件系统 HDFS 作为默认测试数据集存储文件系统。

5.1.2.1 集群规模

集群规模选择的原则是保证 Spark 可执行的最小集群数，将测试压力充分反映在计算节点的资源上，并最大程度降低不必要的网络 I/O 引入的额外资源开销，从而准确衡量 DPU 加速 Spark 的性能。

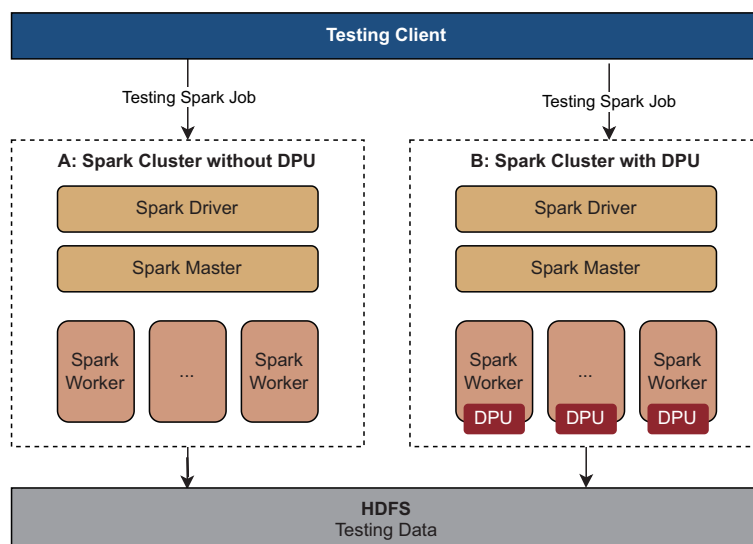


图 5.2: Spark 测试系统示意图

5.1.3 Benchmark

5.1.3.1 测试数据集

测试数据集选择的原则应真实模拟实际业务场景。基于 DPU 加速的 Spark 的一个典型应用案例是 OLAP (On-Line Analytical Processing)，所以以 OLAP 作为测试基准。数据集选择标准基准程序 TPC-DS 中提供的针对大数据应用场景的测试数据。

TPC-DS 模拟了一个真实的商品零售业务场景，数据采用雪花模型，包含 7 张数据表，测试数据大小可以根据实际需求设置为不同大小，例如 1TB、3TB、10TB、30TB 及 100TB。

5.1.3.2 数据量 (data size)

数据量是指使用 TPC-DS 的数据生成工具生成的数据量规模，大数据测试程序生成的数据量以 TB 量级为单位，以支持 HDFS 的数据格式作为数据文件的标准格式。参考

数据量参数为 1TB 或 10TB，参考数据格式为 arquet 格式。

5.1.3.3 测试指标

吞吐量 (throughput) Spark 对海量数据处理，衡量其性能的关键指标为数据处理的吞吐量，即在指定 Spark 任务和数据集下，Spark 每秒可处理的数据表。

处理延时 (latency) Spark 的数据处理分为 Map 阶段和 Reduce 阶段。Map 和 Reduce 处理本身是计算密集型操作；而 Map 和 Reduce 之间的过程为 shuffle, shuffle 会引起大量的网络 IO 和磁盘 IO，此为 IO 密集型。所以，收集各阶段的处理时延，可准确的衡量 DPU 加速效果在不同阶段的差异，即 DPU 分别在计算、存储和网络层面对 Spark 加速的影响。

5.1.3.4 Spark 加速测试 (C101.1sparkbench)

输入： 数据，Query 语句

输出： 单条 Query 语句多次执行的平均时间

测试工具： TPC-DS Benchmark

测试描述： 基于 Spark 的的数据运算复杂性体现在对海量数据的搬运以及查询算子的执行效率。其性能指标体现在数据的处理时延和吞吐量。关于不同计算架构或计算引擎在处理大数据上的性能测试，TPC 委员会提出了一系列被业内广泛认可的标准化测试集。这里我们采用基于 TPC-DS 的测试集进行基于 DPU 的 Spark 性能测试。TPC-DS 会使用一个真实的商品零售业务场景，数据采用雪花模型，包含 7 张数据表，测试数据大小包含多种大小，1TB，3TB，10TB，30TB，100TB。

5.2 AI 加速

人工智能 (Artificial Intelligence, AI) 探索人类的学习模式，并产生一种以人类智能相似的方式做出反应的智能机器。影响 AI 计算性能的关键因素在于算法的执行效率。随着 AI 算法和相关技术的不断发展，AI 适用的领域愈加广泛，实现计算所需的资源不

断增加。实现 AI 计算通常依赖专用的硬件加速设备，如 GPU，TPU 等。但是，AI 模型的结构通常是针对某一任务定制的，不同场景下输入数据规模、模型规模以及算力需求差异明显，这就要求专用硬件架构具有一定的泛化能力。另一方面，影响 AI 计算性能的关键因素在输入数据的获取速率，或称为数据带宽。AI 算法的数据一般要求以规则的格式或特定的形式输入 AI 专用硬件设备以完成计算，数据带宽决定了专用计算引擎的计算效率上限。AI 计算的数据可能直接存储在本地，也可能需要通过网络接收远端数据，且数据清洗预处理过程通常由本地 CPU 执行。相较于高吞吐量的 AI 专用加速器，数据的传输、预处理过程是限制其性能的瓶颈。

5.2.1 基于 DPU 的 AI 加速结构

DPU 加速 AI 计算的基本结构如图 5.3 所示。一方面，DPU 将数据接收、数据预处理等 AI 计算的前序步骤实现硬件卸载，释放了主机端 CPU 的计算负荷，降低 CPU 的资源消耗，同时利用高效的网络卸载引擎和专用数据处理引擎，加速数据接收和处理，降低 AI 计算的前序数据获取开销。另一方面，利用专用的 AI 计算引擎可以进一步加速 AI 计算。DPU 定制化的硬件结构能够提升 AI 计算的全流程计算性能。

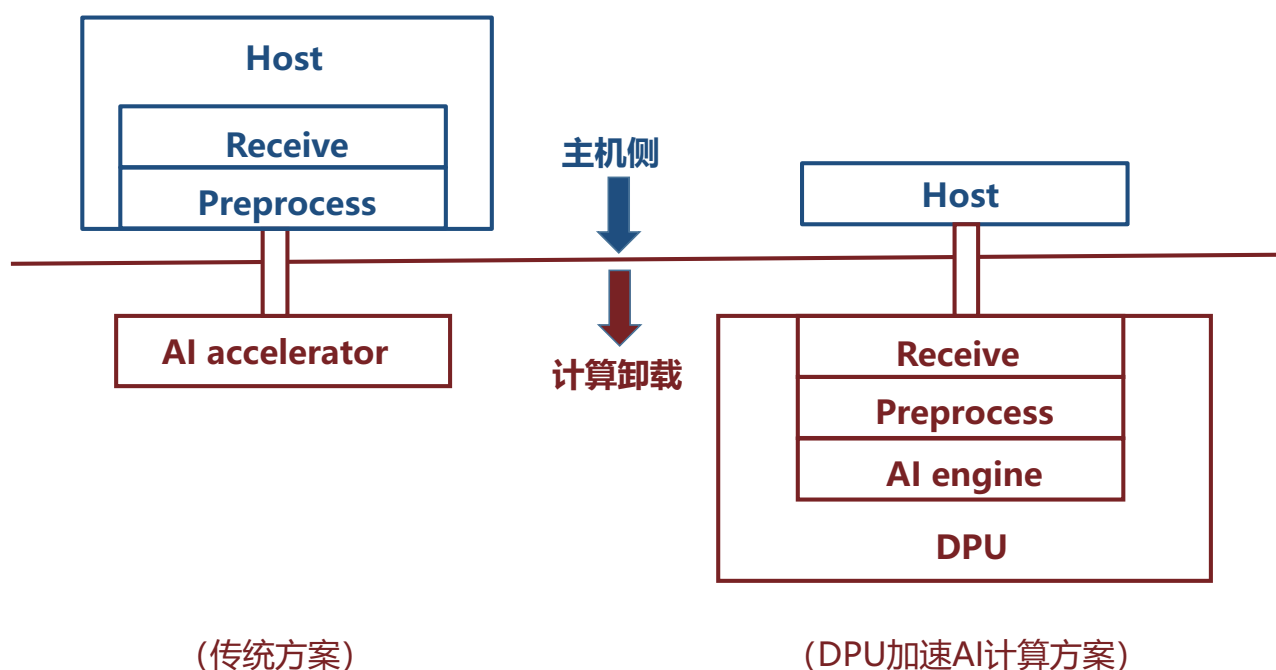


图 5.3: DPU AI 加速结构

5.2.2 测试系统

测试 DPU 的 AI 加速能力可以根据测试场景的不同，采用单端型测试系统或端到端型测试系统两种模式，如图5.4所示。单端型测试系统可认为是一个离线的计算节点，可以用于测量 DPU 加速 AI 计算的峰值性能。端到端型测试系统除中，DPU 除了计算卸载外，还执行网络中的数据接收、预处理，实现 AI 计算在实际复杂应用场景中的全流程业务。

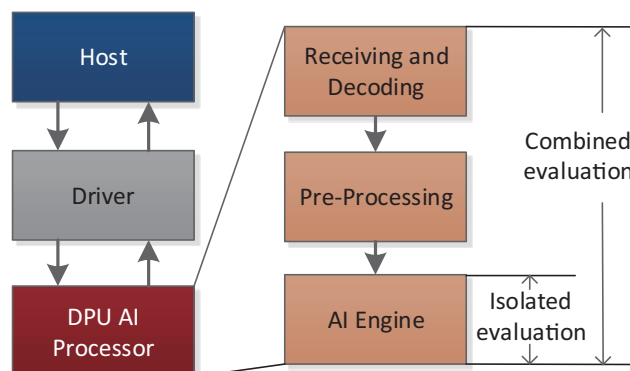


图 5.4: DPU AI 加速测试系统

单端型测试系统将测试程序所需的数据提前储存在 DPU 或主机内存中，根据应用场景选取神经网络模型并加载其权重，完成 AI 计算的测试并统计性能指标。端到端型测试系统将原始数据存放在 initiator 端，target 端利用配置的 DPU 完成数据接收、预处理和计算三个步骤，并统计性能指标。

5.2.3 Benchmark

5.2.3.1 数据集负载参数

DPU 可以实现不同业务场景中的 AI 计算。当 DPU 中的 AI 引擎收到计算指令时，需根据实际需求加载测试数据完成 AI 计算的性能评测。测试数据集的选择可以根据实际应用需求的差别选取现有常用的 AI 标准测试数据集。例如图像识别应用领域的基准数据集 ImageNet，视频分割应用领域的基准数据集 UCF101 等。相应的数据集负载参数根据数据集特征而有差别。但至少应包含以下几个基本类别：

输入数据大小 (Data Size) 用于描述一次计算所需的输入数据的大小或批量输入数据的大小，是描述测试数据集的数据量的基本指标，单位为 Byte，或 KB 等。

输入数据精度 (Data Precision) 用于描述待测数据集数据的精度,如 FP32,FP16,INT8,INT4。

期望结果 (Result) 用于描述待测数据集经过 AI 计算系统后的输出期望结果,该指标根据实际应用需求,可能为分类组别、分类组数等不同具体参数形式。

5.2.3.2 神经网络模型负载参数

一般来说,DPU 可以针对不同业务场景中的 AI 计算进行硬件卸载。这些计算模型或算法的选择与应用场景关系密切。不同 AI 算法适用的计算场景有所区别,例如卷积神经网络适用于视频、图像的识别,Self-attention 算法、隐马尔可夫模型适用于语音识别。但这些模型的量化可以归因于以下几个指标:

数据精度 (Data Precision) 数据精度,如 FP32,FP16,INT8,INT4,表示 DPU 中数据运算和存储所用精度。为节约神经网络模型所占用的存储空间,降低存储权重精度是一个有效的办法,但这种方式会造成识别精度下降。

参数数量 (Parameter Size) 神经网络模型中所包含的参数数据量。

存储空间 (Storage space) 训练完毕的神经网络权重集所占据的空间,和参数量以及参数储存精度相关。

模型计算量 (Computational Complexity) AI 计算执行一次所需的浮点数操作次数或定点操作次数,单位为 FLOPs 或 OPs。

5.2.3.3 测试负载模型

参考 MLperf 基准测试程序,本文给出几种常见应用场景的测试数据以及算法网络模型的参考,如表5.1所示。具体应用场景和 AI 计算可选用其他通用的基准测试数据集。

5.2.3.4 测试指标

模型加载延迟 (Model Loading Latency) 收到执行 AI 计算任务后,DPU 加载神经网络模型权重所需时间。

表 5.1: AI 测试负载模型

场景	数据集	网络模型
图像分类	ImageNet	ResNet-50
图像分类	ImageNet	MobileNet-v1
物体检测	COCO	SSD (ResNet backbone)
物体检测	COCO	SSD (MobileNet backbone)
智能翻译	WMT16 EN-DE	GNMT

数据预处理延迟 (Data Preprocess Latency) 从数据接收到 DPU 后开始计时，至完成数据准备结束。该指标是一个复合指标，用于评测 DPU 在系统中执行网络数据接收及接收数据后执行预处理的总延迟。

神经网络模型计算时间 (Computing time) DPU 完成一次 AI 计算任务所需时间。

能量消耗 (Power Consumption) 用于评测 DPU 执行一次 AI 计算消耗的能量或功率，单位为 J 或 W。该指标也可以根据具体应用定义为单位操作的功耗消耗，例如每幅图像、每像素、每帧所需功耗，单位为 W/pixel，或 W/frame 等。

5.2.3.5 测试场景

表 5.2 记录了本文 benchmark 评估前推计算引擎的五种场景，包括其使用的数据集、参考模型以及基准精度。任务以及网络模型参考 MLPerf Inference Benchmark。

表 5.2: AI 评估场景

任务	参考模型	参数量	数据集 (分辨率)	基准精度
图像分类 (HEAVY)	ResNet-50	25.6M (224×224)	ImageNet	99% of FP32, 76.46% TOP-1
图像分类 (LIGHT)	MobileNet-V1	4.2M	ImageNet (224×224)	99% of FP32, 76.56% TOP-1
物体检测 (HEAVY)	SSD-ResNet-34	36.3M	COCO (1200×1200)	99% of FP32, 76.46% TOP-1
物体检测 (LIGHT)	SSD-RMobileNet-V1	6.91M	COCO (300×300)	99% of FP32, 76.46% TOP-1
机器翻译	GNMT	210M	WMT16EN-DE	99% of FP32, 23.9 SACREBLEU

5.2.3.6 AI 推理加速测试 (C201.1aibench)

输入： 待测试业务的指定数据集。

输出： 该数据集符合精度要求的推理结果，模型加载时间，数据预处理时间，神经网络前推时间，能耗。

测试工具： 根据应用场景选择适配的算法模型。

测试描述： AI 算法以其出众的性能，应用越加广泛。但 AI 推理计算量庞大，对硬件加速的依赖性高，DPU 部署 AI 引擎在 DPU 端完成推理计算，从而避免数据搬运等造成的额外计算开销。在评价 AI 引擎性能方面，学术界和相关行业共同提出的 MLPerf 基准测试给出了较为全面的数据集以及评价方法。DPU AI 推理加速测试在独立评价 DPU 的 AI 加速引擎时，参考 MLPerf 的数据集以及模型；联合评估时，结合网络以及计算模块共同评估。评价指标包括，模型加载时间、推理数据预处理时间以及神经网络模型前推时间。

5.3 数据库加速

数据库按照一定的规律存储数据，并实现数据的查询和修改，通常可以分为关系型数据库和非关系型数据库两类。在关系型数据库中，数据通过关系模型来表示，行或者列的形式进行存储；非关系型数据库（包括键值存储数据库、文档型数据库、图形数据库和时序数据库等），更多应用于更灵活简单的非关系数据模型。DPU 数据库加速针对的主要是关系型数据库。SQL 查询任务在数据库应用中经过解析优化后变为一系列物理算子，数据库加速通过将这其中的数据计算密集型算子卸载到 DPU 上，达到对整体查询性能的提升。

5.3.1 基于 DPU 的数据库加速结构

基于 DPU 的数据库加速结构如图 5.5 所示。在运行数据库系统的节点上安装 DPU 设备，在执行数据库查询操作时，将数据密集型计算算子（如扫描、连接、聚合等）卸载到 DPU 中，从而加速对数据库查询操作。

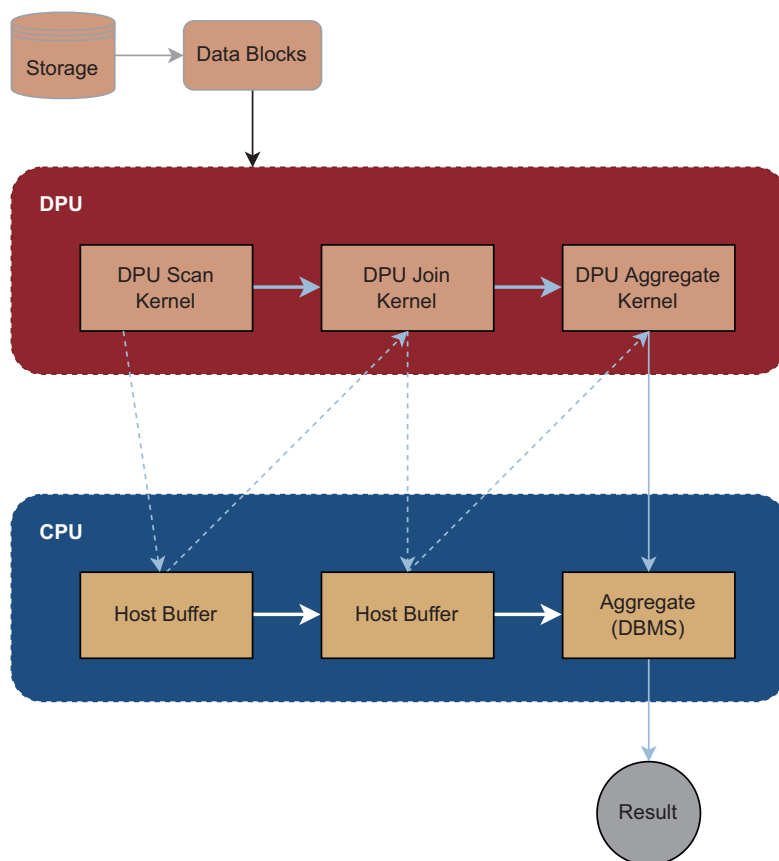


图 5.5: 数据库加速结构

5.3.2 测试系统

对于数据库应用，DPU 数据库基准测试测评 DPU 对单计算节点的加速性能，由此采用单端型测试系统。数据库的性能主要取决于数据吞吐量和处理时延，其中处理时延主要包括数据加载时延 (Load Time) 和查询时延 (Query Time) 两个维度。

如图5.6，测试系统由 A、B 两台具有相同软硬配置的计算节点组成。其中，A 作为比较基准，B 额外增设 DPU 及软件配置，包括驱动和数据库中对 DPU 加速支持的扩展模块或插件。

5.3.3 Benchmark

5.3.3.1 测试数据集

DPU 数据库测试基准可根据不同数据库计算业务需求，选择数据库基准测试程序 TPC 中不同测试子集。

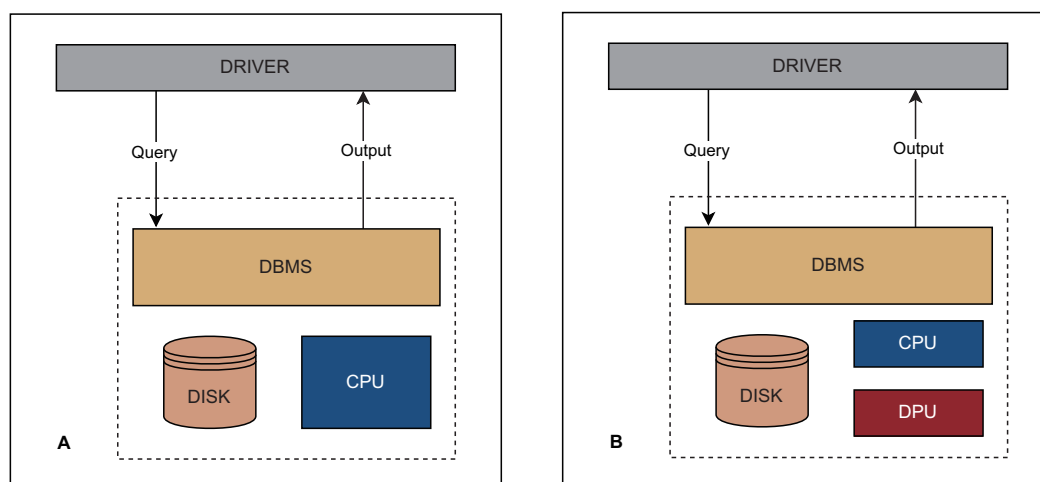


图 5.6: 数据库加速测试系统

5.3.3.2 测试负载参数

数据库大小(database size) 数据库大小是指测试时数据规模,以 GB 为单位($1\text{GB}=2^{32}\text{Byte}$),参考取值为 1GB, 10GB, 100GB, 1000GB 等,每次以 10 倍递增。也可以根据数据量规模选取更大的数据量纲,如 TB 等。

5.3.3.3 测试指标

吞吐量 (throughput) 吞吐量为衡量一个数据库系统性能的重要指标之一,通常以 Qph@Size(Query-per-hour@Size) 为单位,描述在 Size 指定的数据规模下,系统每小时可处理事务数。

处理延时 (latency) 处理时延为衡量一个数据库系统性能的另一重要指标。处理时延由两部分组成:**数据加载时延 (load time)** 是指被测试系统在测试准备阶段,将测试数据从本地文件载入数据库所需要的时间;**查询时延 (query time)** 是指测试准备完成后,被测试系统运行所有待执行查询语句的总时长。

5.3.3.4 数据库加速测试 (C301.1doebench)

输入:

1. 由 TPC 用例 (可以是 TPC-H、TPC-BB、TPC-DS 等不同场景的测试用例) 工具包提供的 DBGen 生成的指定大小的数据集;

2. 由 TPC 用例工具包提供的 QGen 生成的查询语句。

输出：

1. 由 TPC 用例定义的公式计算得出的吞吐量 Qph@Size；
2. 数据加载时延（Load Time）和查询时延（Query Time）。

测试工具： TPC Benchmarks。

测试描述： 关系型数据库的计算复杂性体现在对海量数据的搬运以及查询算子的执行，包括扫描、过滤、投影等。数据吞吐量和处理时延是衡量数据库性能的两个重要指标。DPU 通过数据库算子的卸载，实现对查询计算性能的提升。对性能指标的衡量，我们采用由 TPC 委员会提供的标准数据库测试方法和配套工具，针对事务处理、大数据、决策支持、可视化等多种应用场景下，评估 DPU 加速效果及性能。

5.4 视频/图像编解码

视频业务，包括直播、点播、短视频、视频网站等多种应用形态，已拥有庞大的用户群体且持续增长。数据中心的视频/图像编解码已经成为数据中心对外输出视频业务服务的核心计算负载。为减轻网络和存储的压力，视频/图像编解码标准的压缩率随视频分辨率和帧率的增加而不断提升，但以牺牲高昂的算力为代价。有关研究表明，每一代视频编码标准的压缩率提高 50%，计算复杂度提高 10 倍。特别是在一些实时应用场景中，视频/图像编解码对算力要求高，端设备无法满足算力需求，因此视频/图像的编解码工作通常由数据中心完成，然后分发给用户。DPU 作为数据中心中的新型主力芯片，可以将视频/图像编解码卸载至 DPU 专用引擎，以提高视频/图像编解码效率，并且可以降低数据中心的功耗和边际成本。

5.4.1 基于 DPU 的视频/图像编解码加速结构

DPU 实现视频/图像处理的硬件卸载主要针对视频/图像转码中的解码、图像处理和编码三类计算密集型任务。DPU 加速视频/图像编解码 (Codec) 的加速结构如图5.7所示，计算核包含三部分：解码核（Decoder Core）、图像处理引擎（Image Filter）和（编码核Encoder Core）。此外 DPU 具有强大的网络、存储协议栈卸载功能，除视频/图像编解码计算部分，DPU 还可以利用网络卸载部分接口如 RDMA 和 NOE（Network Offloading

Engine, 网络卸载引擎, 包括对 TCP 和 UDP 两种网络传输层协议的卸载加速), 进一步降低处理延时、提高吞吐量。

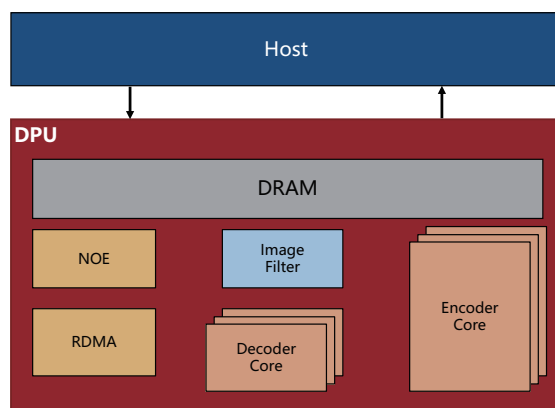


图 5.7: DPU 编解码加速结构

5.4.2 测试系统

DPU 视频/图像编解码基准测试系统是一种端到端型测试系统, 包含主机端和设备端, 其结构如图 5.8 所示。主机端将视频通过驱动发送给 Transcoder 并配置相关编码参数, 在编码完成后将视频流取回; Transcoder 主要包含三个部分: 解码器 (Decoder)、图像处理引擎 (Image Filter) 和 (编码核 Encoder), 视频编解码任务在 DPU Transcoder 进行, 用于对比基于 CPU 的传统编解码计算。

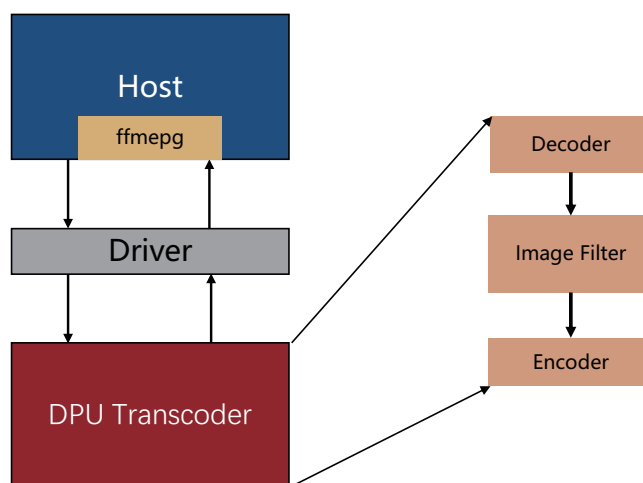


图 5.8: DPU 视频/图像编解码测试系统

5.4.3 Benchmark

5.4.3.1 视频数据集

分辨率 (resolution) 视频帧由矩形像素阵列排列而成，矩阵的行列数为视频的（空间）分辨率，不同设备支持的分辨率不同，视频也可以转码成不同的分辨率，参考分辨率参数有 720p(1280*720)、1080p(1920*1080) 及 4K(3840*2160)。

帧率 (frame rate) 帧率是指视频每秒播放的帧数。通常，帧率越高视频流畅性越好，参考帧率参数有 30fps(frame per second)、60fps 及 10fps。

视频内容 (video content) 不同的视频内容会影响编解码的质量，如电商直播中视频背景基本不变，而游戏直播中视频背景变化多样；因此需要根据视频内容挑选合适的视频组成相应的视频数据集，如 vbench¹从 YouTube 大规模视频集种挑出 15 个具有代表性的视频，如表 5.3所示：

表 5.3: vbench 视频数据集

Video Name	FPS	Resolution
Cat	20	854×480
Holi	30	854×480
Desktop	30	1280×720
Bike	29	1280×720
Cricket	30	1280×720
Game2	30	1280×720
girl	30	1280×720
Game3	59	1280×720
Presentation	25	1920×1080
Funny	30	1920×1080
House	30	1920×1080
game1	60	1920×1080
Landscape	29	1920×1080
Hall	29	1920×1080
chicken	30	3840×2160

5.4.3.2 图像数据集

分辨率 (resolution) 图像帧由矩形像素阵列排列而成，图像的行列数成为图像的（空间）分辨率，不同设备支持的分辨率不同，图像也可以转码成不同的分辨率，常见的分

¹参考链接: <http://arcade.cs.columbia.edu/vbench/>

分辨率有 720p(1280*720)、1080p(1920*1080) 以及 4K(3840*2160)。

格式 (image format) 图像包含不同类的格式，如 JPEG、PNG、WebP、HEIC 等等。

图像内容 (image content) 即图像的生成源，如 WebP 定义有 photo、picture、drawing、icon、text 几类。

5.4.3.3 配置参数

视频编解码标准 (video codec standard) 常见的视频编解码标准有 H.264、H.265、VP-9 以及 AV1, 我国自己的标准有 AVS1、AVS2 以及 AVS3, 转码过程中需要根据输入视频的编码格式以及输出视频的编码格式要求进行配置 Transcoder。

图像编码标准 (image codec standard) 常见的图像编码标准有 JPEG、WebP、PNG、GIF 和 HEIF 等，由于不同设备 and 应用支持的图像编码格式不同，经常会将图像转码成多个标准。

编码参数 (encoding parameter) 视频/图像编码过程中，根据相应的质量约束需要对视频/图像编码器的编码参数如量化参数、关键帧比率进行配置，常见约束有常数 QP、码率约束以及质量约束。编码器会根据相应约束自适应调整编码参数。

量化参数 (Quantization Parameter, QP) 量化是编码中一个非常重要的环节，是将连续值映射到多个离散值的过程从而实现压缩，但这一过程会引入失真，因此量化同时影响视频质量和码率，量化参数 QP 就是量化的关键参数，在 H.265 中取值范围为 0~51，在 WebP 中取值范围为 0~100。在编码过程中，可以固定 QP 值也可以自适应调节 QP。

关键帧比率 (key frame ratio) 视频帧分为 I 帧、B 帧和 P 帧，I 帧为关键帧进行帧内预测，B 帧和 P 帧主要进行帧间预测，关键帧的比率会影响视频流的质量，关键帧比率增大，视频失真度会减小，但视频压缩率会降低。同时关键帧的比率也会影响吞吐量和能效比等参数。

5.4.3.4 测试指标

吞吐量 (throughput) 衡量转码器单位时间可以转码的像素点个数，单位为 Mpixels/s。视频在不同分辨率下像素点数不同，根据吞吐量这一指标就可以评估不同分辨率和不同帧率的视频转码延时，吞吐量越大转码延时越小。

响应时间 (response time) 即端到端的服务响应延时。通常用 P95、P99 等指标来衡量响应时间。

能效比 (power efficiency) 衡量转码器单位功耗可以转码的像素点个数，单位为 Mpixels/s/J。不同转码器转码视频时功耗不同，利用能效比这一参数可以直观反映 DPU Transcoder 的能耗。

质量 (quality) 不同的 Transcoder 加速器在设计时候会在编解码算法上做一定的近似，这一近似增加了加速器在吞吐量和能效比的表现，但是会影响质量，质量包含失真和压缩效率两部分，失真通常用 PSNR(peak signal-to-noise rate，以 dB 为单位) 衡量，压缩效率反应在视频的码率 (bit rate，以 kbps 为单位)，通常会将两个指标结合在一起，得到一条速率失真 (RD) 曲线，并利用 BD-rate (Bjontegaard Delta-Rate) 评估编码器的整体性能。

5.4.3.5 视频/图像编解码加速测试 (C401.1codecbench)

输入：

1. 视频数据集 (由标准测试集 vbench 和网络随机视频数据集构成)。
2. 图像数据集 (由标准测试图 Lenna、Tecnick 和网络随机图像数据集构成)。
3. 视频格式，包括原视频格式参数、目标视频参数格式，参数包含分辨率、帧率、量化参数和关键帧比率。
4. 图像格式，包括原图像参数格式、目标图像参数格式，参数包含分辨率、量化参数。

输出： 利用 ffmpeg 测试不同视频/图像数据集、不同视频/图像格式编解码要求下的吞吐量、响应时间、和质量。

测试工具： ffmpeg。

测试描述： DPU 可以完成不同视频/图像标准、格式的编解码加速需求，将 DPU 编解码加速引擎驱动层集成到标准的视频应用软件如 ffmpeg，利用 Benchmark 中的标准数据集和参数集合作为输入，可以准确测试 DPU Transcoder 的性能指标。

5.5 云原生加速

云原生 (Cloud Native) 作为云计算新一代发展范式，塑造了业务应用开发，部署，维护的新的形式。服务网格(Service Mesh) 是构建云原生生态的关键技术，它以解耦的方式为业务应用提供了基础设施服务，构建了一个专用且透明的中间平台，使业务应用的开发可以专注业务本身。服务网格的组织架构通常包括一个集中式的控制平面和一组代理构成的分布式数据平面。控制平面负责整个网格系统的维护，例如配置下发，状态检测等等。数据平面则作为网格数据处理的核心由多个代理构成。这些代理与服务部署在一起也被称为边车代理 (sidecar proxy)，代理所有进出服务的流量，实现流量管控，安全检测，数据统计等功能。作为数据处理的主要部件，这些代理在实现强大网络功能的同时也为整个云原生平台引入了巨大的开销，例如延时增加，吞吐降低，CPU 占用严重等。为解决这些问题，DPU 可以通过软硬协同的数据处理方式加速服务网格的整个数据平面。

5.5.1 基于 DPU 的云原生加速结构

DPU 作为一个 I/O 数据处理加速平台，提供了丰富的软硬件功能，为加速服务网格提供了必须的硬件前提。同时，DPU 能够加速服务网格的另一个重要原因在于两者在功能上定位的相似性，它们都是处理数据进入服务前的预处理，如网络协议解析，加解密，序列化和反序列化等等。除此之外，DPU 提供的软硬件可编程能力也将在服务网格的加速中发挥重要作用。硬件可编程将使其具有一定的可扩展性，能够满足服务网格新的变化的需求，而由高性能 ARM 核提供的软件可编程能力则能够将相关的控制功能卸载到该平面，避免控制信息在 Host 和 DPU 之间的冗余传递。通过二者的结合，DPU 将利用硬件加速的优势赋能服务网格为云原生平台提供一个低延迟，高带宽的服务通信平台。

由 DPU 构建服务网格的数据平面对其性能的改善主要在两方面。一方面是通过硬

件加速计算密集型操作降低延时，增加吞吐，例如路由转发，加密解密，压缩解压缩等。另一方面是通过同节点边缘代理与边车代理的融合优化数据平面整体架构，消除冗余路径。同时，通过将服务网格数据平面卸载到 DPU 上也减少了对主机端 CPU，内存资源的消耗。

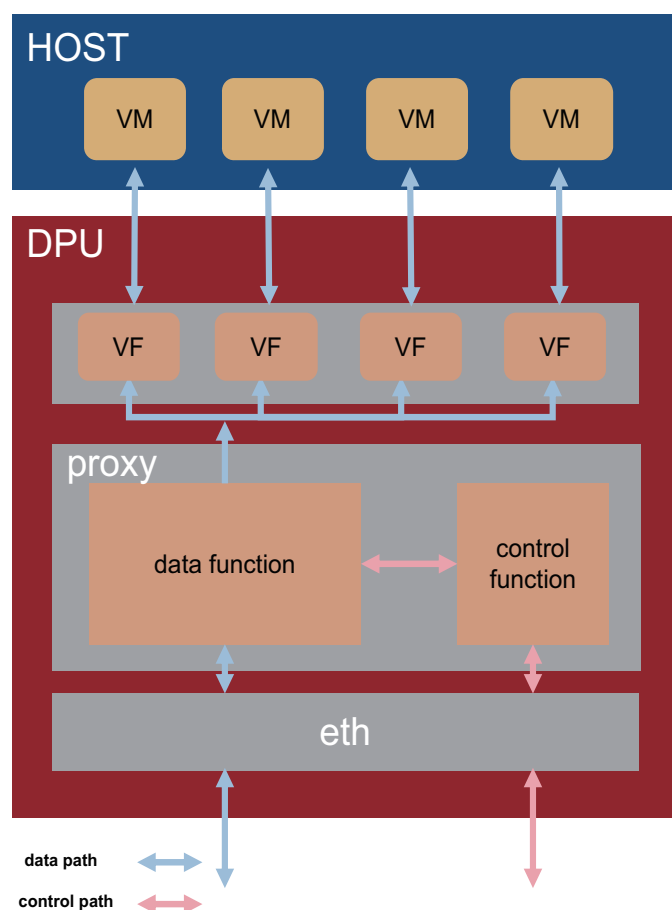


图 5.9: DPU 加速服务网格架构

如图 5.9，主机侧的业务应用运行在虚拟机 (VM) 或容器中，通过 VF 与实现在 DPU 上的代理通信。DPU 实现的代理利用硬件实现数据处理功能完成 I/O 数据的预处理，利用 ARM 核实现控制功能与服务网格控制平面进行通信协商。

5.5.2 测试系统

服务网络的测试依赖于上层服务的通信。在云原生中服务的通信方式通常包括同一节点内的服务通信以及跨节点的服务通信，所以测试分为两类，单端节点内服务通信测试（单端型测试系统）和对端节点间服务通信（端到端型测试系统）。对于该测试系统分为运行服务的 benchmark 层和实现服务网格加速的 DPU 层。具体介绍如下：

1. 基准测试层 (Benchmark): 该层作为数据的起始和终止位置, 提供不同类型的服务对服务网格的不同网络层次处理能力进行验证。这些服务提供各类负载, 如 TCP 数据包, UDP 数据包, HTTP 请求等。同时, 也能够根据测试需求和相关负载参数的要求提供不同的测试数据。
2. DPU 层: 该层对于测试来说是黑盒, 由方案提供商完成基于 DPU 的服务网格加速。其需要完成服务网格的相关功能, 同时保持与现有的上层云平台的正常交互。

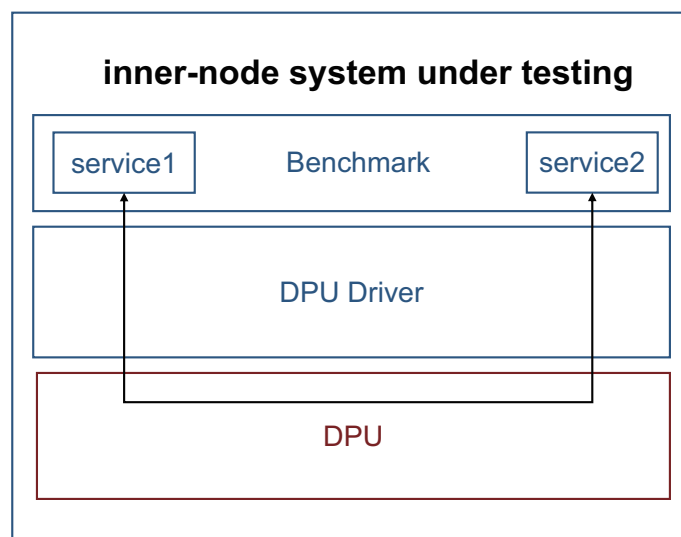


图 5.10: 服务网格单端型测试系统

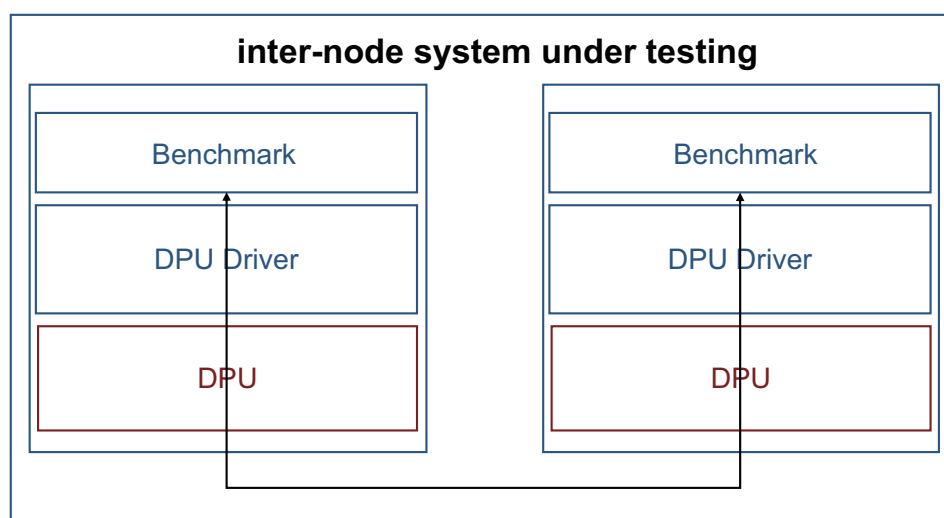


图 5.11: 服务网格端到端型测试系统

通过该系统测试, 我们希望了解不同负载对基于 DPU 实现的服务网格的影响。不同负载包括, 数据包的大小, 链接的数量, 服务 (service) 的数量等。了解在不同功能需

求下基于 DPU 实现的服务网格的性能变化。不同功能需求包括，仅 L3/L4 层处理，同时需要 L3/L4 和 L7 层处理等。

5.5.3 Benchmark

5.5.3.1 L3/L4 服务测试负载

- **数据包大小 (packet size)** 对于 L3/L4 的延迟测试，通常需要利用小包测试性能峰值，包容量参考区间为 [16,32K]Bytes。超过 MTU 的包会被切片测量总的延时。
- **流量大小 (Traffic size)** 测试 L3/L4 的吞吐通常需要通过变化流量大小来看其变化趋势。参考取值范围是 128KBytes-16GBytes。
- **平台复杂度 (platform complexity)** 平台复杂度通常是指同时工作的 service 以及链接的数量，使用 ovs 的转发规则可间接表示平台的复杂度。参考取值范围为 1-1000000 条。

5.5.3.2 L7 服务测试负载

- **每秒请求数 (QPS)**: L7 层数据测试通常关心请求与响应次数。其负载通常是发送的请求数,取值范围可以是 1-100000QPS。
- **链接数量与服务数量 (number of connection and service)**: 高并发是 L7 层请求处理的一个非常重要的场景。在测试服务网格性能的时候也需要通过变化链接数量和服务数量去看它的影响。

5.5.3.3 负载模型

依据不同服务类型，服务网格性能的测试分为 L3/L4, L7 两类。L3/L4 处理的数据为 raw data，其关心的是包的大小。L7 处理的数据是文本，其关心的是请求的数量。同时，我们也需要考虑平台复杂度带来的影响，其中 L3/L4 的直接影响条件是OVS的转发规则数量，L7 是连接的数量。下面我们定义相关的负载模型作为后续测试用例基本的输入要求。

服务网格延时测试负载模型 在类似飞行控制这类实时性场景中，延时是评价系统的重要指标。该负载模型5.4主要对类似场景产生的数据进行建模，测试小包数据的延时指标。

表 5.4: 服务网格延时测试负载模型

测试名称	包大小 (Bytes)	ovs 转发规则数量
包处理延时 1	16	1
包处理延时 2	64	1
包处理延时 3	256	1
包处理延时 4	1024	1
包处理延时 5	16	1K
包处理延时 6	16	5K
包处理延时 7	16	10K
包处理延时 8	16	25K
包处理延时 9	16	50K
包处理延时 10	16	100K

服务网格吞吐测试负载模型 在数据中心对吞吐量有一定的要求。该负载模型5.5 即希望通过模拟不同流量的大小，测试 DPU 数据吞吐量。

表 5.5: 服务网格吞吐测试负载模型

测试名称	流量大小 (Bytes)	ovs 转发规则数量
流量处理吞吐 1	128KB	1
流量处理吞吐 2	512KB	1
流量处理吞吐 3	2MB	1
流量处理吞吐 4	8MB	1
流量处理吞吐 5	32MB	1
流量处理吞吐 6	128MB	1
流量处理吞吐 7	512MB	1
流量处理吞吐 8	2GB	1
流量处理吞吐 9	8GB	1
流量处理吞吐 10	16GB	1K
流量处理吞吐 11	16GB	5K
流量处理吞吐 12	16GB	10K
流量处理吞吐 13	16GB	25K
流量处理吞吐 14	16GB	50K
流量处理吞吐 15	16GB	100K

服务网格应用层测试负载模型 在服务网格中，L7 层主要的通信协议是 Http。对于该层的测试，我们主要关注的 QPS（Query Per Second）的情况，connection 的数量，用于负载均衡的服务数量。因此，该负载模型5.6主要模拟 L7 层发送请求的情况。

表 5.6: 服务网格应用层测试负载模型

测试名称	qps	请求端建立的连接数	services
http 请求处理 1	1	1	100
http 请求处理 2	10	1	100
http 请求处理 3	100	1	100
http 请求处理 4	1000	1	100
http 请求处理 5	10000	1	100
http 请求处理 6	100000	1	100
http 请求处理 7	1000	2	100
http 请求处理 8	1000	4	100
http 请求处理 9	1000	8	100
http 请求处理 10	1000	16	100
http 请求处理 11	1000	32	100
http 请求处理 12	1000	64	100

5.5.3.4 测试指标

服务网格的目的是为上层的 services 提供通信服务与 services 本身的逻辑无关。所以，评估的关键在于能否为 service 提供更好的通信能力，理论上评估应该包括功能和性能两部分。其中，功能的评估主要是支持功能有哪些；性能的评估则是在其提供必要功能的基础上评估这些功能的性能，而不关心服务 (service) 本身的性能怎样。

对于服务网格来说它的一切功能都是围绕服务之间的通信来展开的，所以我们主要关注的性能指标与网络类似，包括延时，吞吐量。另外，在云场景下各类资源的高效利用是降低成本的重要方式，故对于 CPU 的有效使用率也应该作为一个重要指标。

综上，我们确定了评估硬件实现服务网格数据平面的关键指标，CPU 利用率，延迟和吞吐。同时，服务网格实现多层网络代理的特点使得在不同层次的测试中具有些许的差别。下面我们对各层的指标进行详细介绍。

L3/L4 服务测试指标

- **延时 (Latency):** 对 L3/L4 层数据来说，延时的衡量包括单程数据传输的延时以及一个来回的 RTT。
- **吞吐量 (Throughput):** 从 L3/L4 层的角度看，数据传输是 bit 流的形式，故其吞吐通常由单位时间 bit 数表示。

L7 服务测试指标

- **百分比延迟 (Percentile latency)** L7 层的延迟尤其要考虑长尾延时，故测量需要更

全面的表示。通常，延时测试中会根据响应数量的百分比统计延时，例如 50% 延时，70% 延时，90% 延时以及 99% 延时等。

- **每秒请求数 (QPS)** L7 层的数据传输通常以请求响应方式进行，在衡量其吞吐时通常使用单位时间成功响应的请求数。

5.5.3.5 云原生网络传输层数据传输测试 (C601.1cloudnativebench)

输入： 小包延时测试负载模型模拟了网络中不同大小包发送的情形；吞吐测试负载模型模拟了网络中不同大小流量发送的情形；

输出： 通过延时测试获得不同大小包处理的延时信息，同时获取 CPU 的占用情形。通过吞吐测试获得不同大小流量情形下系统的吞吐量信息，同时获得 CPU 的占用情形；

测试工具： Iperf3 和 sockperf；

测试描述： 应用服务有直接基于网络传输层网络协议进行数据通信的，例如 MongoDB, Postgresql 等应用。对于这些应用，服务网格主要是在 L3/L4 层发挥作用，我们进行测试就需要验证这个层次的性能情况。

5.5.3.6 云原生网络应用层数据传输测试 (C602.1cloudnativebench)

输入： 应用层负载模型模拟了不同 QPS 以及不同并发的网络情形；

输出： 利用 Fortio 工具模拟应用层负载，测试百分比延时，成功响应 QPS 以及 CPU 占用信息；

测试工具： Fortio；

测试描述： 在现阶段的微服务架构中，不同服务之间的通信往往借助应用层协议，例如 http1.1, http2.0, grpc 等等。这些协议处于网络应用层能够提供更多用于管控的信息，但同时也带来的一系列性能问题。为解决这些问题，DPU 提供了一个很好的硬件加速平台。因此，在测试服务网格在 DPU 上应用实现的性能时，这是一个极其重要的测试场景。

第 6 章 面向安全的基准评测

信息的爆炸式增长、网络规模的不断扩展，安全成为各计算系统和设备必须具备的特征。在现代信息系统中，硬件设备安全及芯片安全是处理器设计必须具备的能力。作为一种新型的数据处理器，DPU 提供的安全能力分为两部分，第一部分是处理器自身的安全，如采用物理不可克隆函数（PUF）技术；第二部分是对系统整体的数据安全处理及与安全需求相关的技术（安全技术领域包括应用层、网络层、传输层等多个层次）支持。DPU Benchmark 安全部分着重测试 DPU 对系统提供的安全能力（即上述第二点）。系统安全能力包括：DPU 通过增加对网络报文进一步的（L4~L7）解析与规则匹配（如 DPI、IPS/IDS），将无效流量（恶意流量）拦截在 DPU 一侧，降低对主机端上行通路的带宽需求及 CPU 处理时间的消耗；通过将安全协议中计算复杂度较高的运算过程（如 IPsec 的加密算法和验证算法等）交由 DPU 处理，加速协议执行过程。DPU Benchmark 安全部分主要表征 DPU 安全加速的性能。

6.1 DPI

DPI（Deep Packet Inspection，深度包检测）是一种对网络数据流量应用层内容进行深度检测识别的机制，可以基于内容识别结果对网络流量进行安全控制、镜像、统计和流日志等行为处理，广泛对应用于运营商网络和企业安全网络中。

DPI 主要包括规则库、策略、统计和日志等功能模块。DPI 设备通过对报文解析，识别 2~4 层报文头信息；再辅助深度内容检测，完成基于内容级别的特征字符匹配，识别出更细分的业务流量；并根据对应流量类型的策略，完成转发、丢弃等基本数据处理行为，以及统计、镜像和流日志等更高级的业务运维统计。进一步的，这些业务运维统计数据，可以作为业务分析、网络流量优化等重要参考依据。

6.1.1 基于 DPU 的 DPI 加速架构

传统的 DPI 实现方法是基于软件实现的。软件实现 DPI 在协议解析、识别等业务处理中，计算效率低。利用 DPU 上原生的可编程的协议解析和识别功能单元，硬件实现 DPI 的处理效率远高于 CPU 上的软件方式。基于 DPU 的 DPI 加速结构如图 6.1 所示。将 DPI 任务执行过程中最为耗时的协议解析和协议识别卸载到 DPU 中，并且对于协议

处理的 Action 进行硬件实现，从而起到对 DPI 安全引擎任务的加速作用，实现更高效的 DPI 处理方法。

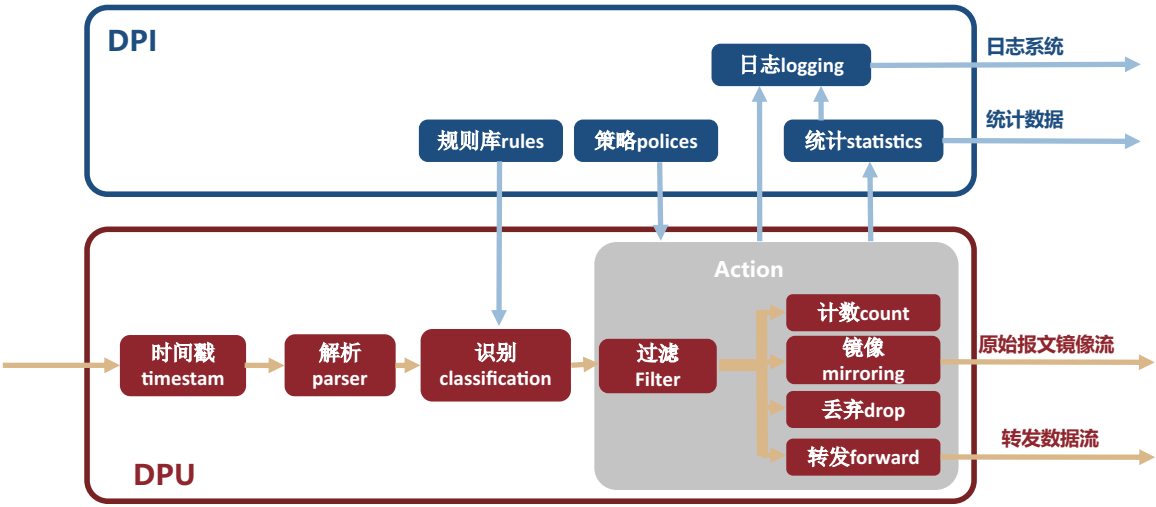


图 6.1: DPI 加速结构

6.1.2 测试系统

DPU 的 DPI 测试系统采用端到端型测试系统，结构如图6.2所示，测试模拟服务器 (server) 在不断生成被测设备服务器 (server) 的输入流量。通过不同特征负载作为输入，测量 DPU 的 DPI 加速能力，重点关注 DPU 加速后的性能指标，如识别延时、吞吐量和时延变化等。

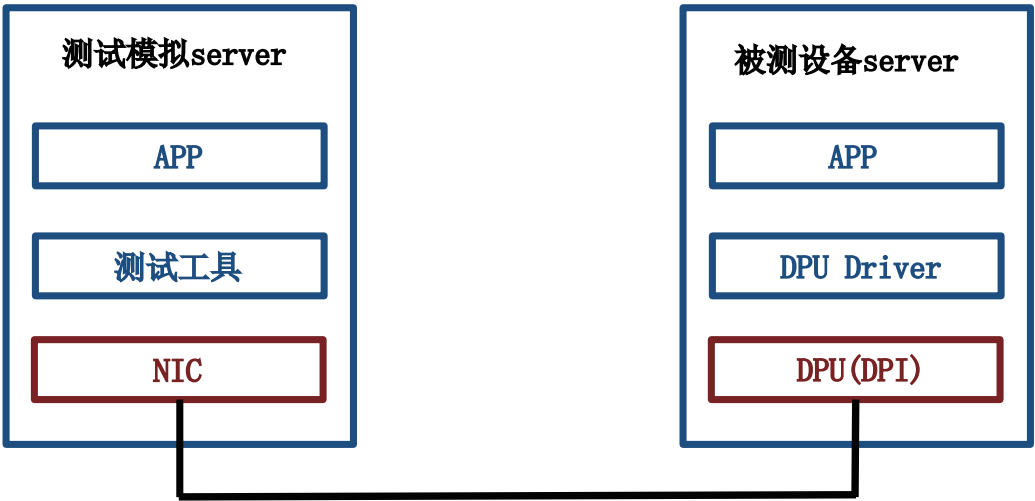


图 6.2: DPI 测试系统

6.1.3 Benchmark

6.1.3.1 测试负载参数

包大小 (packet size) 包大小是最关键的参数，影响 DPU 包处理的吞吐量和速率，单位为 Byte。

包速率 (packet rate) 发包速率，表示每秒发送包的个数。

包类型 (packet type) 侦测数据包类型，来判断业务和应用类型。不同的应用类型体现在会话连接或数据流上的状态各有不同。

6.1.3.2 测试指标

误报率 (false alarm rate) 误报率是指某种类型的网络业务流量被误识别为其它类型网络业务流量在所有被测试网络业务流量样本中的占比。

漏判率 (miss rate) 漏判率是指网络业务流量中预期应该被识别出来的业务类型没有识别到的网络业务流量占总网络业务流量样本的百分比。

识别率 (recognition rate) 识别率是指能够识别的网络业务流量在总网络业务流量样本中的占比。识别率作为 DPI 系统中的指标值，具有参考意义，因为识别率内的数据还可能存在识别误报的情况。

识别准确率 (identification accuracy) 识别准确率是指准确识别的网络业务流量在总网络业务流量样本中的占比。流量识别的准确率是体现 DPI 能力的最关键指标之一。

识别延时 (latency) 识别延时是指从流量样本进入 DPI 设备到 DPI 设备识别出网络业务流量类型所消耗的时间。识别延时是体现流量识别性能的重要指标之一，也是 DPU 能够加速提升性能的重要指标。

吞吐量 (throughput) 吞吐量是 bytes per second，该指标 DPI 的整体包处理能力，包括正常转发和丢弃的。

抖动 (jitter) 时延变化主要是每个包在 DPI 处理中时间的抖动 (最大延时与最小延时的差值)。

6.1.3.3 DPI 加速测试 (S101.1dpibench)

输入： 根据测试负载分类生成的不同类型不同大小的数据包，来评估 DPI 的过滤规则和检测能力。

输出： 根据两种测试场景的测试用例生成测试日志记录，包括如下核心测试指标，包类型识别准确率、识别误判率、识别漏判率、识别率、识别准确率、识别延时、识别延迟、时延变化、吞吐量。

测试工具： Perf。

测试描述： 通过测试仪表或测试工具根据测试负载分类模拟各种类型的测试流量，通过测量各种类型的测试负载在被测试设备上表现的行为特征，来判断基于 DPU 的 DPI 在性能指标方面的能力和优势。

6.2 防火墙

防火墙是现代数据中心网络和企业网络的重要安全设备之一，作为网络入口，能够有效的防御常见的网络攻击。防火墙通过配置安全规则实现对特定服务的流量放行及对非法流量的阻拦。随着云计算业务的繁荣发展，传统的硬件防火墙无法满足云计算数据中心内安全基础设施的弹性伸缩能力的要求，进而衍生出虚拟防火墙，通过对防火墙虚拟化，实现按需的部署和删除，满足不同用户业务流量级别的规模需要。

6.2.1 基于 DPU 的防火墙加速架构

基于 X86 的软件方式实现防火墙，需要大量 CPU 资源来处理业务流量类型解析和匹配等复杂业务逻辑，CPU 执行的低能效比严重增加了数据中心运营成本。基于 DPU 对防火墙做硬件加速，可以利用其可编程能力实现防火墙功能卸载，通过 in-line 的方式解析网络数据包并进行防火墙规则匹配，能够有效降低 CPU 资源使用率，实现更小空间、更高能效比的可弹性伸缩的防火墙解决方案。

如图6.3，通过将防火墙的控制平面和数据平面卸载到 DPU 设备，将防护墙的数据转发丢弃规则以及有状态会话的状态卸载到 DPU 中，从而实现基于 DPU 卸载加速的防火墙应用。基于 DPU 的防火墙加速方案可以结合云计算VPC 网络能力实现基于租户的防火墙策略，拥有更广泛灵活的应用空间。

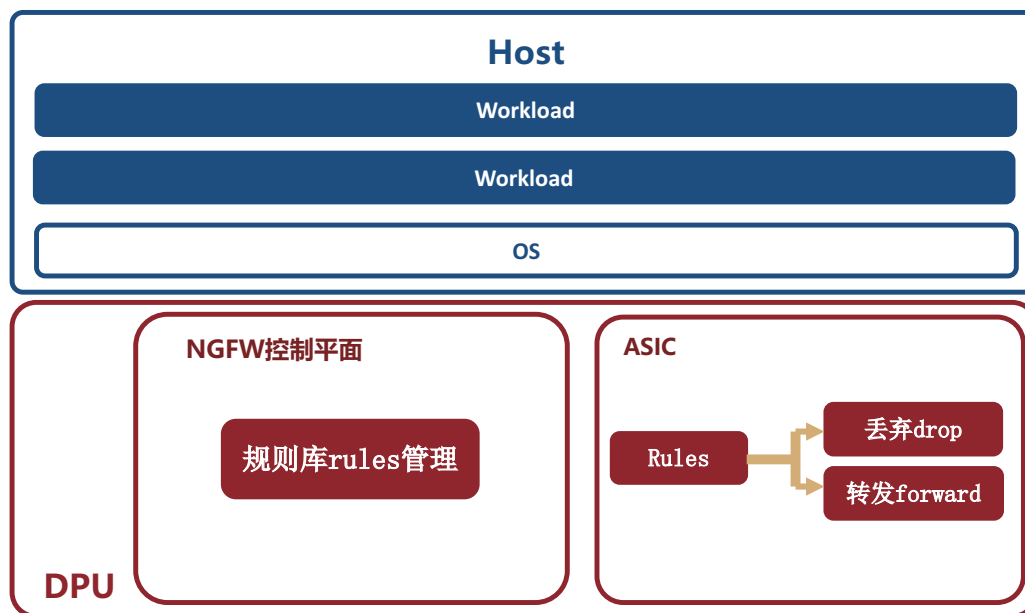


图 6.3: 基于 DPU 的防火墙加速架构

6.2.2 测试系统

基于 DPU 的防火墙测试系统采用端到端型测试系统。安全功能产品部署在数据中心流量的主要路径上，DPU 防火墙基准测试关注转发性能及整体网络的吞吐量和时延。DPU 防火墙基准测试系统包括两个服务器 (server)，其中一个 server 上部署测试工具比如 MiniSMB，负责发起测试报文，另一个服务器部署 DPU，并且该 DPU 具备防火墙能力，该服务器还可以部署模拟 APP，用来测试 DPU 的应用连接 TCP/HTTP 建立能力，结构如图6.4所示。

6.2.3 Benchmark

6.2.3.1 测试负载参数

包大小 (packet size) 包大小指数据包长度，不同数据包长度的报文在防火墙处理转发过程中所消耗的时间不同。

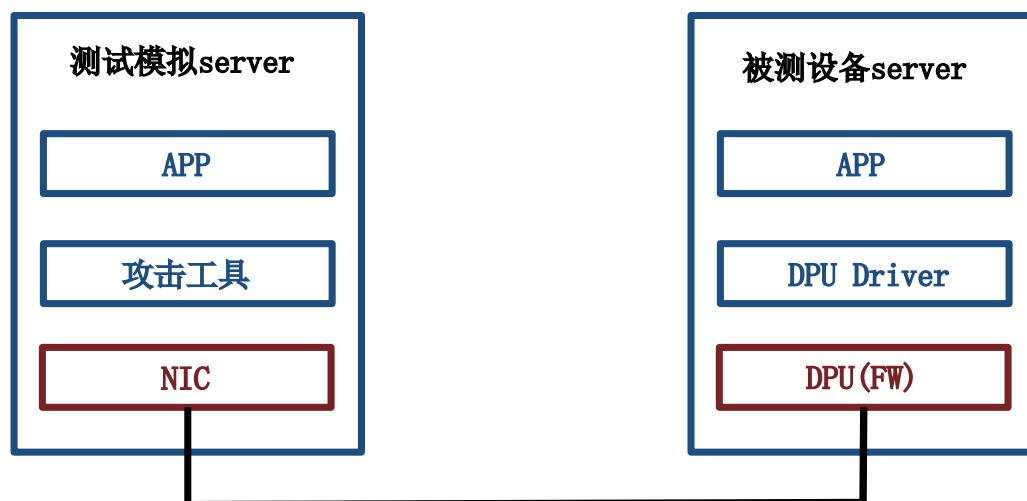


图 6.4: 防火墙测试系统

包速率 (packet rate) 通过调节发送包速率参数来对防火墙的吞吐能力进行检测，通常在不丢包情况下防火墙能处理的最大包速率能力是防火墙设备的吞吐能力的重要指标。

包类型 (packet type) 防火墙包过滤支持很多种报文封装类型，比如 TCP，UDP，IP 等。对于 DPU 来说每种报文封装类型都有不同的处理时间消耗，所以 DPU 防火墙功能测试需要把报文封装类型作为基础参数。

6.2.3.2 测试指标

吞吐量 (throughput) 吞吐量是在各种帧长的满负载双向发送和接收数据包而没有丢失情况下的最大数据传输速率，代表 DPU 的整体包处理能力。

延时 (latency) 延时代表从测试数据帧的最后一个比特进入 DPU 处理至测试数据包的第一个比特从 DPU 离开的时间间隔。

最大会话数量 (maximum number of session) 最大会话数量指基于 DPU 防火墙所能顺利建立和保持的最大并发 TCP/UDP 会话数。

每秒新建连接数 (Connection Per Second,CPS) 每秒新建连接数指一秒以内基于 DPU 防火墙所能建立及保持的 TCP/UDP 新建连接请求的数量。

6.2.3.3 防火墙加速测试 (S201.1firewallbench)

输入： 根据测试负载分类，配置 Ftester（具体配置方法参考 Ftester 使用手册），并模拟生成测试负载数据。

输出： 根据测试生成的日志文件，来评估防火墙的过滤规则，生成的测试日志文件都将输出最大、最小、平均延迟，取平均延迟；取最大并发连接数与每秒新建连接数输出。

测试工具： Ftester。

测试描述： 通过测试仪表或测试工具根据测试负载分类模拟生成测试负载数据，测试负载数据经过被测试设备后，记录被测试设备表现出来的性能特征，根据性能测试判断被测试设备经过 DPU 加速后的性能提升效果。

6.3 IPS/IDS

入侵防御系统（Intrusion Prevention System, IPS）/入侵检测系统（Intrusion Detection System, IDS）都是采用入侵检测技术来防御和识别网络威胁的常见网络设备。IPS/IDS 对流经网络路径上的数据流进行 4~7 层的深度分析识别，实现精确、实时地识别并阻断或限制常见网络攻击行为，进而有效预警并保护网络。

6.3.1 基于 DPU 的 IPS/IDS 加速架构

传统的 IPS/IDS 产品基于 X86 CPU 上软件方式实现，需要大量 CPU 资源来解析和识别网络威胁，其网络吞吐量和时延性能不足以满足现有需求。基于 DPU 的 IPS/IDS 加速架构通常利用硬件实现的正则匹配（RegEx）专用加速器实现特征检测功能的加速，能够实现高性能的特征分析。结合 DPU 具备的连接状态追踪能力，可以对有状态协议状态的管理跟踪进行加速。基于 DPU 的 IPS/IDS 加速架构结构如图6.5所示。

6.3.2 测试系统

基于 DPU 的 IPS/IDS 测试系统通常采用端到端型测试系统。DPU IPS/IDS 基准测试系统包括两个服务器 (server)，其中一个服务器上部署测试工具如 Snort，负责发起攻

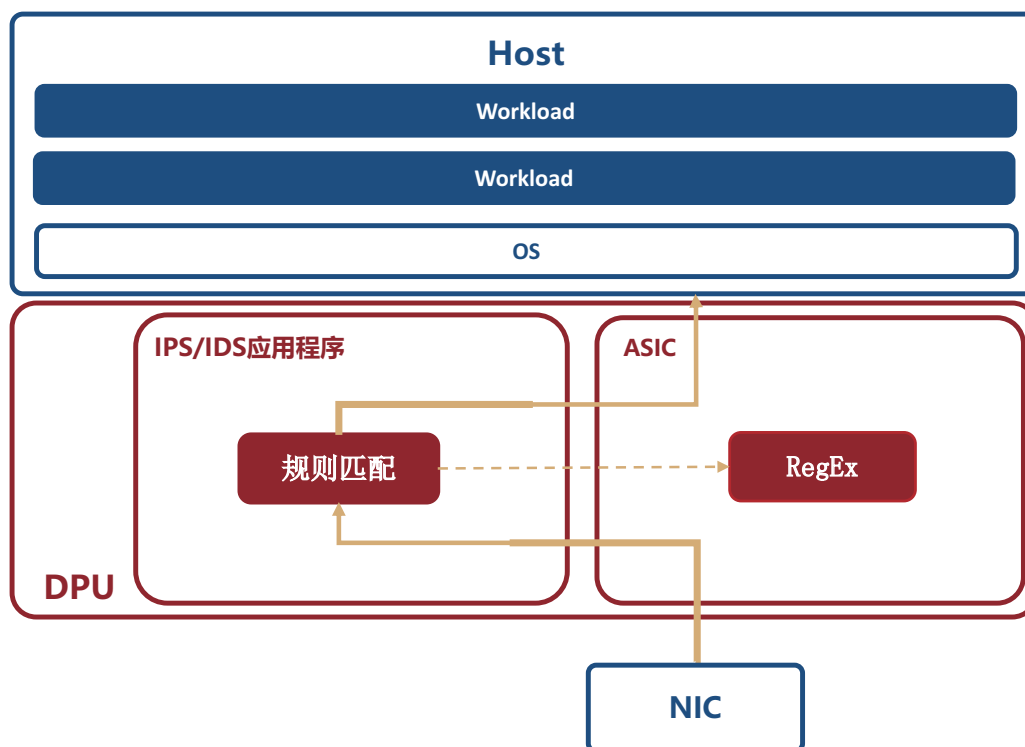


图 6.5: IPS/IDS 加速结构

击测试报文，另一个服务器上部署 DPU，且该 DPU 具备 IPS/IDS 能力，该服务器还可以部署模拟 APP，用来测试 DPU 的应用连接 TCP/HTTP 建立能力，具体如图6.6。

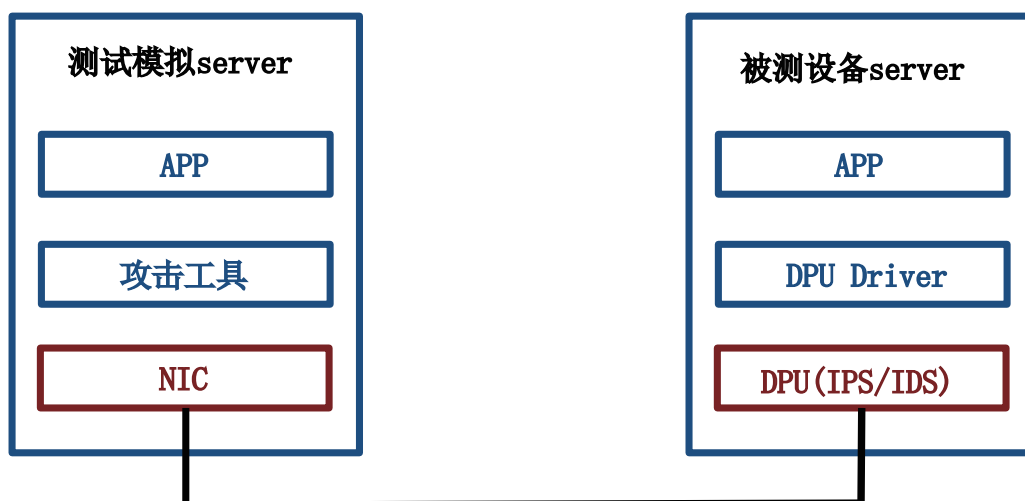


图 6.6: IPS/IDS 测试系统

6.3.3 Benchmark

6.3.3.1 测试负载参数

包大小 (packet size) 包大小是指报文长度，不同长度的报文对于 IPS/IDS 特征识别效率有较大影响。

包速率 (packet rate) 通过调节发送包速率参数来对 IPS/IDS 的吞吐能力进行检测，通常在不无故丢包情况下 IPS/IDS 能处理的最大包速率能力是 IPS/IDS 设备的吞吐能力的重要指标。

6.3.3.2 测试指标

吞吐量 (throughput) 吞吐量是在各种帧长的满负载双向发送和接收数据包而没有丢失情况下的最大数据传输速率，代表 DPU 的包处理能力，一般在处理 64Byte 小包的时候，吞吐量会有所下降。

延时 (latency) 延时代表从网络业务流量进入被测试设备到被转出或者丢弃所消耗的时间。

误报率 (false alarm rate) 误报率作为 IPS/IDS 的重要指标，是指测试用的网络业务流量样本中被错误识别的比例。

漏判率 (miss rate) 漏判率是指测试用的网络业务流量样本中没有被识别到的比例。

识别准确率 (identification accuracy) 识别准确率是指测试用的网络业务流量样本中被准确识别的比例。流量识别的准确率是体现 IPS/IDS 能力的最关键指标之一。

6.3.3.3 IPS/IDS 加速测试 (S301.1ipsidsbench)

输入： 通过 Fttester 按测试负载分类进行配置，然后生成的对应的测试负载数据流量，作为测试输入数据。

输出： 根据测试生成的日志文件，来评估 IPS/IDS 的检测能力，生成测试日志文件都将输出最大、最小、平均延迟，取平均延迟；取吞吐量和识别率来评估 IPS/IDS 的检测能力。

测试工具： Ftester。

测试描述： 通过测试仪表或者测试工具按照测试负载类型模拟测试数据，包括正常数据和攻击数据，然后经过被测试设备进行识别和转发，记录对应的测试结果，包括行为和性能指标数据，评估经过 DPU 加速后的 IPS/IDS 性能表现。

6.4 IPSec

IPSec 是 IP Security 的简称，是 IETF 制定的隧道加密认证技术，为 IP 层应用负载提供高安全性保障。IPSec 协议中涉及对 IP 负载进行加密、解密以及认证等算法操作。其计算复杂度较高，采用软件方式实现的 IPSec 会占用大量主机 CPU 资源。同时由于单个 CPU 核处理能力有限，而一个 IPSec 会话通常会在一个指定 CPU 核上处理，所以，IPSec 的吞吐能力也是 CPU 上软件实现方案瓶颈。通过 DPU 实现在线卸载 IPSec 协议，对数据包进行实时封装和解封装，可以有效提高 IPSec 单会话的吞吐能力，并有效降低 CPU 负载。

6.4.1 基于 DPU 的 IPSec 加速结构

基于 DPU 的 IPSec 隧道加速通常利用 DPU 上加解密引擎，对 IPSec 数据进行实时在线加解密，同时 IPSec 协议再通过 DPU 硬件进行封装解封装，进而实现 IPSec 隧道的加速。如果 IPSec 用于云计算中，通常还会与云平台虚拟交换机（如 OVS）一起部署，所以，在硬件流水线中完成完整的数据平面处理，体现了 DPU 芯片对于网络和安全协议融合应用的支持能力。基于 DPU 的 IPSec 架构如图 6.7 所示，OVS 控制平面和 IPSec 控制平面运行在 DPU 的 SoC 上，对 ASIC 上的数据平面统一管理配置。硬件 ASIC 上不只实现 OVS 数据面流表，还能够在对应的 Action 中支持 IPSec 的加解密和封装解封装。



图 6.7: 基于 DPU 的 IPSec 隧道加速示意图

6.4.2 测试系统

基于 DPU 的 IPSec 隧道测试系统通常采用端到端型测试系统，其结构如图6.8所示。IPSec 隧道测试系统能够模拟建立多条 IPSec 隧道会话，以及在 IPSec 隧道内部模拟多条连接会话。两侧主机分别接收发送会话数据，测量基于 DPU 卸载的 IPSec 吞吐能力和时延指标等，用于评估 DPU 在 IPSec 应用领域的能力。

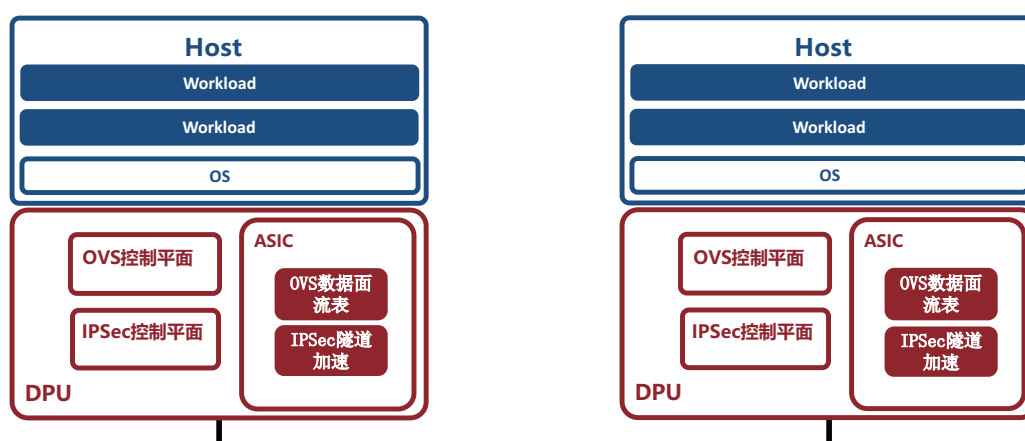


图 6.8: 基于 DPU 的 IPSec 隧道测试系统

6.4.3 Benchmark

6.4.3.1 测试负载参数

包大小 (packet size) 包大小是最关键的参数，影响 DPU 包处理的吞吐量和速率。包大小以 Byte 为单位。参考取值范围 [96,9000] Bytes。理论上包越大，DPU 的吞吐量越大。

包速率 (packet rate) 发包速率，该参数代表测试工具的每秒发送包的个数。

IPSec 隧道会话并发创建 通过测试仪表模拟 IPSec 的隧道会话，衡量 DPU 上 IPSec 隧道会话支持能力。

6.4.3.2 测试指标

吞吐量 (throughput) 吞吐量是在各种帧长的满负载双向发送和接收数据包经过 IPSec 隧道的最大数据传输速率，代表 DPU 上 IPSec 的整体包处理能力。包括 PPS (Packet Per Second) 指标和 BPS (Bits Per Second) 指标。

时延 (latency) 时延代表从测试数据报文进入基于 DPU 到转发出去所消耗的时间间隔，象征着 DPU 上 IPSec 的数据处理性能。

抖动 (jitter) 抖动是指通过 DPU 上 IPSec 转发数据的时延波动情况，抖动越大代表 DPU 处理 IPSec 数据转发过程中性能越不稳定。

IPSec 隧道会话数量 (maximum number of IPSec tunnels) 最大 IPSec 隧道会话数量指 DPU 上 IPSec 隧道会话所能顺利建立和保持的最大并发会话数。

每秒新建 IPSec 会话数 (Connection Per Second, CPS) 每秒新建连接数指一秒以内 DPU 上 IPSec 所能建立及保持的 IPSec 隧道会话的数量。

IPSec 单会话最大吞吐量 (throughput) 代表一条 IPSec 隧道会话在不丢包的情况下所能收发处理的最大吞吐量。

6.4.3.3 IPSec 硬件加速性能测试 (S401.lipsecbench)

输入： IPSec 报文、正常 TCP/UDP 通信报文；

输出： 整体吞吐量、单会话吞吐量、时延、抖动、建连速度；

测试工具： 测试仪表（或 Iperf3）；

测试描述： 首先配置 DPU 上 IPSec 相关参数，通过 DPU 的背靠背配置方式，在 DPU 所在主机系统上使用 Iperf 批量建立连接会话，使得不同会话触发建立不同的 IPSec 隧道，进而测量 IPSec 隧道建立速度等性能指标，通过 Iperf 还能够测量整体吞吐能力、时延和抖动等性能指标。

第7章 总结

DPU 作为新近发展的专用数据处理器，不同厂商的 DPU 功能差异化显著，有的侧重网络可编程，有的侧重存储卸载，有的侧重安全与虚拟化。对于用户而言，不仅对 DPU 的功能定位有误差，也无法获取 DPU 真正的性能指标。同时，由于 DPU 新的功能定位，如远程资源本地化（DPU 将承担网络协议栈的处理，对主机端屏蔽网络细节）。而且 DPU 的某些主要功能的评测没有借鉴，需要重新定义评测标准。本文提出的 DPU 性能基准评测根据“POC 原则”（真实性、针对性、全面性），围绕网络、存储、计算与安全四个维度的应用场景，引入各自典型应用作为基准测试，测试 DPU 的相关指标。

DPU 在不同的应用场景中几乎都改变了传统的应用结构，通过释放主机端 CPU 资源，负责处理数据执行过程，起“降本增效”的作用。在每个基准测试中，首先介绍该应用基于 DPU 的加速结构；其次，根据第二章提出的三种测试系统模型（单端型，端到端型和多端型）构建实际测试系统模型；最后，定义在这种应用场景中 DPU 测试负载的关键参数，测试指标以及测试描述。在测试负载与指标的选取过程中，有的充分参考借鉴了已有领域的测试基准，如 Spark 加速、数据库加速采用 TPC Benchmarks，这些是业界公认测试标准，我们认为，DPU 作为底层加速器，采用已有的 Benchmark，仍然能表征 DPU 的性能；有的根据应用的负载参数，定义了负载模型，模拟真实场景，如 NVMe-oF 加速测试、云原生服务网格加速测试。

DPU 性能基准评测的提出，首先对于用户而言，一方面获得 DPU 功能定位，另一方面获得性能参考；其次对于 DPU 厂商来说，通过对 DPU 产品基准评测结果，可以不断优化自己的设计；最后对于研究者而言，也能够提供对研究 DPU 的帮助。DPU 性能基准评测的使命是为未来的 DPU 产品建立一套公平、开放、全面、客观的 DPU 评测体系，引导 DPU 产品标准化。

DPU 性能基准评测中，测试工具目前采用开源工具，对于某些复杂场景以及特殊测试需求，如 SD-WAN、5G MEC 等，简单的测试工具无法或者测试系统无法反映 DPU 在其中的性能，需要进一步完善。

DPU 支持的场景非常丰富而且还在不断拓宽，DPU 性能基准评测仅选取当前典型的应用作为主要的评测方法，需要根据 DPU 功能的变化增加或修改，不断完善 DPU 性能基准评测方法。同时，根据本文提出的 DPU 性能基准评测方法，我们未来也将对成熟的 DPU 产品进行测试，并给出测试结果及报告。

附录 A Benchmark 汇总

表 A.1: Benchmark 汇总表

Benchmark	描述
N101.1basicnetbench	内核态 TCP/UDP 端到端延迟测试
N102.1basicnetbench	内核态 TCP/UDP 发送带宽测试
N103.1basicnetbench	内核态 TCP/UDP 接收带宽测试
N201.1rdmabench	RDMA 端到端延迟测试
N202.1rdmabench	RDMA 端到端带宽测试
N203.1rdmabench	RDMA QP 并发能力测试
N301.1ovsbench	OVS 加速测试
N401.1sdwanbench	SD-WAN 加速测试
N501.1lbbench	LB 加速测试
N601.15gmecebench	5G MEC 加速测试
M101.1nvmeofbench	裸盘性能测试
M102.1nvmeofbench	虚拟化存储性能测试
C101.1sparkbench	Spark 加速测试
C201.1aibench	AI 推理加速测试
C301.1doebench	数据库加速测试
C401.1codecbench	视频/图像编解码加速测试
C601.1cloudnativebench	云原生网络传输层数据传输测试
C602.1cloudnativebench	云原生网络应用层数据传输测试
S101.1dpibench	DPI 加速测试
S201.1firewallbench	防火墙加速测试
S301.1ipsidsbench	IPS/IDS 加速测试
S401.1ipsecbench	IPSec 硬件加速性能测试

附录 B 测试工具

B.1 Netperf

Netperf 是由惠普公司研发并开放的网络性能测量工具，根据应用的不同，可以进行两种模式的网络性能测试，包括批量数据传输（bulk data transfer）模式和请求/应答（request/reponse）模式，支持 TCP 和 UDP 传输层协议。Netperf 工具以 client/server 方式工作。server 端是 netserver，用来侦听来自 client 端的连接，client 端是 netperf，用来向 server 发起网络测试。在 client 与 server 之间，首先建立一个控制连接，传递有关测试配置的信息，以及测试的结果：在控制连接建立并传递了测试配置信息以后，client 与 server 之间会再建立一个测试连接，进行来回传递特殊的流量模式，以测试网络的性能。

B.2 Iperf3

Iperf3 是一个广泛使用的网络性能测量工具，是一个跨平台的工具，支持 Windows、Mac OS X、Linux、FreeBSD 等各平台使用。Iperf3 可以测试最大 TCP 和 UDP 带宽性能，根据测试需要可以调整多种参数和 UDP 特性；可以报告带宽、延迟、抖动和丢包率等参数；Iperf3 具有客户端和服务端功能，可以创建数据流来测量两端之间单向或双向的网络吞吐量等相关参数。

B.3 Perftest

Perftest 是一个基于 RDMA Verbs 接口开发的开源 RDMA 性能测试工具，可以对支持 RDMA 技术的设备进行带宽和时延测试。Perftest 在无上下文切换的情况下通过读取 CPU 周期计数器确定时间戳，计时精度高。Perftest 使用往返延迟的一半作为待测 RDMA 操作的单路延迟，适用于对称拓扑。

B.4 Perf

perf 是 Linux 下的一款性能分析工具，能够进行函数级与指令级的热点查找。通过 Perf，应用程序可以利用 PMU、tracepoint 和内核中的计数器来进行性能统计。它不但可以分析制定应用程序的性能问题（per thread），也可以用来分析内核的性能问题，可以同时分析应用程序和内核，从而全面理解应用程序中的性能瓶颈。

B.5 TPC Benchmarks

由 TPC 委员会提供的数据库性能测试方法和配套工具。TPC 已有成熟的针对事务处理、大数据、决策支持、可视化等多种应用场景下的测试数据，以及配套的测试查询语句 (Query)，包括且不限于 TPC-H 数据及其 22 个测试用例，TPC-DS 及其 99 个测试用例、TPC-BB 及其 30 个测试用例等。且所有类型的测试数据均提供对不同数据规模的支持，包括 1GB, 10GB, 1TB, 10TB, 100TB。

B.6 FIO

FIO 是一个测试裸盘性能的工具。它可以产生多线程或进程来执行用户指定的特定类型的存储 I/O 操作，可以模拟多种存储负载模型。FIO 支持的读写模式包括顺序读，随机读，顺序写，随机写，混合随机读写，混合顺序读写。常用参数包括引擎，队列深度，线程，block，是否裸设备，读写方式，大小/耗时，跳过缓存等

B.7 Imagemagick

Imagemagick 是创建、编辑、撰写、转换位图图像的处理工具，可以读入及处理超过 200 种图像格式，同时支持缩放、剪切、调色等多种图像处理功能，将 DPU 图像转码应用层和驱动集成到 Imagemagick 工具中。

B.8 ffmpeg

ffmpeg 是集视频格式转化、视频加水印、视频采集等功能为一体的多媒体视频处理工具，提供了多种媒体格式的封装和解封装等丰富的模块插件，可以将 DPU 视频转码

应用层和驱动集成到 ffmpeg 工具。

B.9 sockperf

sockperf 是一种网络测试工具，旨在测量网络延迟和网络延迟峰值。sockperf 可以创建 UDP/TCP 数据流并测量承载它们的网络的吞吐量和延迟。SockPerf 允许用户定义可用于测试网络的不同参数，或者用于优化或调整网络。同时，该工具提供客户端和服务端功能，并且可以单向或双向测量两个端点之间的吞吐量和延迟。

B.10 fortio

fortio 是一个快速，小型的网络测试工具，能够提供多个层次网络通信的负载测试。它可进行多种延迟结果的统计以及 qps 的相应情况的统计，同时提供了 Web UI 可对测试结果进行可视化。

B.11 Ftester

FTester 全称 Firewall Tester，是一个用来测试防火墙的过滤策略和入侵检测 IPS/IDS 能力的工具。这个工具主要是由两个 perl 的脚本组成：第一部分是 ftest，运行在防火墙或 IDS 的一侧。它根据用户设定的配置文件 ftest.conf，对目标主机进行数据包的注入操作。第二部分是 ftestd，运行在防火墙或 IDS 的另一侧，往往作为 ftest 的目标主机。它负责嗅探注入的各种数据包。通过对比两者的日志文件，来评估防火墙的过滤规则和 IPS/IDS 的检测能力。

索引

A

Artificial Intelligence, AI, 人工智能 6, 49, 52

B

Bits Per Second, BPS, bit 每秒 5

Border Gateway Protocol, BGP, 边界网关协议 31

Business Support System, BSS, 业务支持系统 29

C

Ceph, 分布式文件系统 42

Cloud Native, 云原生 65

Codec, 编解码 60

Connection Per Second,CPS, 每秒新建连接数 36

Connection Per Second, CPS, 连接每秒 5

Connection Tracking, CT, 连接跟踪 28

Control Plane, 控制平面 2

Corner Cases, 长尾场景 3

Customer Premise Equipment, CPE, 用户终端接入设备 30

D

Data Plane, 数据平面 2

Data Process Units, DPU, 专用数据处理器 1, 6

Datacenter tax, 数据中心税 5

Decoder Core, 解码核 60

Deep Packet Inspection, DPI, 深度报文检测 7, 72

Direct Attached Storage, DAS, 直连式存储 41

Direct Routing, DR, 直接路由 35

DPU Benchmark, DPU 基准测试程序 1, 9, 11, 15, 17, 41, 49, 72

E

Encoder Core, 编码核 60

End-to-End, 端到端型测试系统 8, 9

G

GPRS Tunneling Protocol, GTP, GPRS 隧道协议 39

H

Hadoop Distributed File System, HDFS, 分布式文件系统 50

Hypervisor, 管理程序 (虚拟化中) 2

I

IDS, Intrusion Detection System, 入侵检测系统 78

Image Filter, 图像处理核 60

Input/Output Operations Per Second, IOPS, I/O 操作每秒 5

Internet Protocol Security, IPSec, 互联网安全协议 7, 81

Internet Wide Area RDMA Protocol, iWARP 22

IPS, Intrusion Prevention System, 入侵防御系统 78

L

Load Balance, LB, 负载均衡 33

M

Machine Learning, ML, 机器学习 6

MapReduce, 一种编程模型 49

MEC, Multi-access Edge Computing, 移动边缘计算 17

Message Enabler Platform, MEP, 边缘应用平台 37

Mgmt. Plane, 管理平面 2

Multi-access Edge Computing, MEC 36

Multi-End, 多端型测试系统 8, 9

Multi-Protocol Label Switching, MPLS, 多协议标签交换 29

N

Network Address Translation, NAT, 网络地址转换 35

Network Attached Storage, NAS, 网络接入式存储 41

Network Functions Virtualization, NFV, 网络功能虚拟化 4, 30, 37

Network Processor, NP, 网络处理器 43

Non-Volatile Memory Express , NVMe, 非易失性内存主机控制器接口规范	41
Non-Volatile Me, NVM, 非易失性内存	5
NVMe over Fabric, NVMe-oF	5, 13, 41

O

On-Line Analytical Processing, OLAP, 联机事务处理过程	51
Open System Interconnection Reference Model, OSI, 开放式系统互联通信参考模型	34
Open vSwitch, OVS, 虚拟交换机	3, 26, 37, 68, 81
Operation Support System, OSS, 运营支撑系统	29

P

Packets Per Second, PPS, 包每秒	5
Performance、Power、Area, PPA, 性能, 功耗, 面积	1
Physical Function, PF, 物理功能	48
Physical Unclonable Function , PUF, 物理不可克隆函数	72
ping-pong, 乒乓操作	24
Point of Presence, POP	30
Practicality, Orientation, Comprehensiveness, POC, 真实性, 针对性, 全面性	3

Q

Quality of Service, QoS, 服务质量	5, 28
Quantization Parameter, QP, 量化参数	63
Query Per Second, QPS, 每秒请求数	21, 36
Queue Pair, QP, 队列组	24

R

RDMA over Converged Ethernet, RoCE, 融合以太网的 RDMA	22
Redundant Array of Independent Disks, RAID, 独立磁盘冗余磁盘阵列	3
Remote Direct Memory Access, RDMA, 远程直接数据存取	4, 22, 43
Run Time Environment, RTE, 运行时环境	10

S

Service Mesh, 服务网格	65
sidecar proxy, 边车代理	65
Single Root Input/Output Virtualization, SR-IOV	41, 48

Single-End, 单端型测试系统	8, 9
socket, 套接字	18
Software Defined Accelerator, SDA, 软件定义加速器	2
Software Defined Network, SDN, 软件定义网络	1, 37
Software Defined Storage, SDS, 软件定义存储	2
Software Defined Wide Area Network, SD-WAN, 软件定义广域网	29, 32
Spark, 一种用于大数据处理的分布式计算引擎	49
Storage Class Memory, SCM, 存储类内存	41
Storage Performance Development Kit, SPDK	41
System Under Test, SUT, 测试系统	8, 9

T

Transcoder, 转码器	61
Transmission Control Protocol/Internet Protocol, TCP/IP, 传输控制协议/网际协议	3, 18

U

User Plane Function, UPF, 用户平面功能	17, 37
----------------------------------	--------

V

Virtual function, VF, 虚拟功能	48, 66
Virtual Machine, VM 虚拟机	66
Virtual Private Cloud, VPC, 虚拟私有云	4, 76
VLAN, 虚拟局域网	28