DPDK
应用基础

唐宏　柴卓原　任平　王勇　等◎编著

DPDK出色的数据面性能优化能力
使其成为新一代数据平面解决方案

DPDK
APPLICATION BASIS

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

DPDK
应用基础

唐宏　柴卓原　任平　王勇　等◎编著

DPDK出色的数据面性能优化能力
使其成为新一代数据平面解决方案

DPDK
APPLICATION BASIS

人 民 邮 电 出 版 社
北　京

图书在版编目（CIP）数据

DPDK应用基础 / 唐宏等编著. -- 北京 : 人民邮电出版社, 2016.8
ISBN 978-7-115-42604-8

Ⅰ. ①D… Ⅱ. ①唐… Ⅲ. ①应用软件－软件包 Ⅳ. ①TP317

中国版本图书馆CIP数据核字(2016)第192265号

内 容 提 要

本书分为基础原理、DPDK 应用与测试、DPDK 应用开发及实例解析 3 个部分。其中，第一部分简要介绍了DPDK相关背景、技术原理、部分库函数、DPDK安装部署和调试方法，帮助读者从宏观上了解DPDK，对其有一个基本认识。第二部分阐述了 DPDK 技术在 NFV 场景下的应用，并通过 16 个测试用例说明了基于 DPDK 的 NFV 转发性能调优实践与测试结果，以助于业界同仁开展 NFV 系统的研发与性能评估工作。第三部分介绍了基于 DPDK 的应用案例，案例介绍中融合了对如何基于 DPDK 进行上层应用开发的解析，以帮助读者更加深入地理解 DPDK。

本书可以作为网络技术人员和 IT 系统开发人员的中初级读物，帮助他们快速了解 DPDK 社区及其大型通用控制器的开发方法。

◆ 编　　著　唐　宏　柴卓原　任　平　王　勇　等
责任编辑　吴娜达
责任印制　彭志环

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100078　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京隆昌伟业印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：12.5　　2016 年 8 月第 1 版
字数：200 千字　　2016 年 8 月北京第 1 次印刷

定价：49.00 元

读者服务热线：(010) 81055488　印装质量热线：(010) 81055316
反盗版热线：(010) 81055315

前 言

近两年，经过业界孜孜不倦的努力，NFV 技术已取得长足的进步，从以往简单的 PoC 功能验证和初级的标准规范制定，发展到近期在编排、控制与基础设施层有大量的开源项目涌现，如 OpenMANO、ODL、ONOS、OpenStack、OVS 等。同时，基于各类固网/移动电信业务的 VNF 试点已在全球普遍开展。NFV 所需的支撑技术，如 Cloud OS、NFVI 及 VIM 等组件也出现了商用解决方案。另外，NFV 在转发性能方面也取得了长足进步，各类基于 DPDK 技术的单服务器转发能力已经超过 100 Gbit/s，思科最新的 FDio 开源虚拟交换机单服务器（24 核）的混合分组长转发性能已超过 500 Gbit/s，这些都是令人鼓舞的成就。

DPDK 引起我们的关注并非偶然，熟悉电信网络的朋友都知道，移动网的 PGW 网元和宽带网络的 BRAS 类转发设备，都对转发性能有着非常高的要求。如果用 x86 架构的 NFV 服务器取代上述设备，由于 Linux 内核的限制和通用 I/O 报文读写方式，这种基于传统 Linux 和虚拟化技术的转发平面万兆端口普遍仅有 1 Gbit/s 以内的转发能力；而目前的电信级硬件转发设备可以轻松做到 64 byte 小报文不丢失分组线速转发。这种巨大反差的存在，促使我们努力去了解其内在原因，因此，初期我们对 DPDK 技术的研究更多是出于兴趣。

作为来自运营商的研发团队，我们更擅长网络相关的系统集成与测试，很少关注底层实现技术，对现代计算机体系结构的理解也较为粗浅；但 NFV 性能问题又要求团队对高速 I/O 技术及相关的体系结构、操作系统、虚拟化等技术有深度了解和配置能力。在历时 2 年多的时间中，团队克服自身不足与重重困难，逐步积累起 DPDK 在应用与测试方面的使用经验，也与大量网络厂商多次探讨基于 DPDK 的 VNF 加速问题。这些积累也帮助我们理解了 NFV 编程与部署中的并发与核亲和性、巨页与内存管理、NUMA 结构等 NFV 性能的影响因素。DPDK 不仅是一个用户态的 I/O 加速方案，也包含了 NFV 开发测试环境中所需的高速网络应用样例库以及开发调试和测试工具集。在这个过程中，我们欣喜地看到，有越来越多的开源项目或商用解决方案基于 DPDK 进行了实现，如 ODL SFC 项目、OVS DPDK 等，DPDK 已经成为 NFV 加速的基础性架构和公认的应用加速接口。

2015 年底，我们将一部分成果总结为 DPDK 技术白皮书 1.0 版并进行了发布。白皮书包含了我们对 DPDK 技术的理解，介绍了 DPDK 应用场景，基于 DPDK 技术的 NFVI 转发能力测评，这些测试结果是可重现的。很多读者和朋友对白皮书提供了宝贵意见，根据这些建议，我们后续改进了测试方法，增加了海量流表、多网卡并发等测试项，并发布了白皮书

1.2 版。以上内容吸引了很多初学读者和运营商相关同行的关注，他们希望我们能系统地介绍一下 DPDK 入门技术、NFV 转发性能优化技术及其相关的评估方法，这成为本书成稿的主要原因。

本书的阅读对象主要是互联网/电信业从事 SDN/NFV 设计/规划人员、从事网络编程的初中级工程师、NFV 技术集成测试工程师。本书可以作为了解 DPDK 技术和 NFV 性能评测的初级读物，或作为计算机体系结构、Linux 内核技术、虚拟化以及 vSwitch 技术等专业计算机书籍的辅助读物，它就像一条项链，把知识点有机地串接起来，帮助大家理解基本概念及其应用。

本书大致分为 3 部分：第一部分主要介绍 NFV 转发性能问题的成因、NFV 加速和 DPDK 引入的背景、DPDK 原理与架构、主要库函数工具，帮助读者学习 DPDK 的主要技术架构、安装部署、代码结构，了解在开发和测试中可以利用的工具和样例库，该部分内容主要来自 DPDK 白皮书和 DPDK 开源社区提供的文档及代码，因时间有限，我们并未对所有代码进行评测；第二部分介绍了 DPDK 相关的 NFV 性能测试方法学，给出了各个测试例及其测试结果，同时，附录列出了针对不同 OS 版本和 DPDK 版本的详细配置方法和测试条件；本书最后一部分针对开发虚拟化网元的网络编程人员，给出了在 DPDK 驱动、用户态 VNF 代码移植方面的开发示例，如 L3 处理队列、收发逻辑核绑定、收发缓冲区设定等，内容虽然简单，但可以作为代码编写或调试的练习案例。

感谢中国电信广州研究院 SDN 项目团队的各位成员，莫博奇、李鹏、陈前锋、刘其坚等同事为本书提供了大量测试例和测试总结，罗雨佳、刘汉江、欧亮等同事参与了本书部分章节的撰写与修订。同时，也感谢 Intel 公司的 Danny Zhou、朱河清、杨涛、李训、廖阔等专家在 DPDK 评估中提供的大量技术指导，没有他们的开创性工作，我们无法独立胜任这项艰巨的任务。我们还对武汉绿网深圳灵动智网公司提供的大力协助表示诚挚的感谢，尤其是刘江与陈平两位工程师，他们在前期 DPDK 研究测试和本书编程技术的撰写中做出了重要贡献，我们对他们及武汉绿网深圳灵动智网公司广大同事的奉献精神深表敬意。最后，我们真心感谢在 DPDK 测试中为我们无私提供设备和技术指导的各位厂商朋友，希望能与你们继续结伴同行。

衷心希望本书能帮助各位朋友轻松踏上 NFV 之旅，祝大家旅途愉快！

唐 宏

2016 年 5 月于广州

目 录

第一部分 基础原理

第二部分 DPDK 应用与测试

第三部分 DPDK 应用开发及实例解析

附 录

第一部分 基础原理

第一部分简要介绍DPDK的技术背景、开源社区、工作原理、相关库函数以及DPDK的安装部署和调试方法，帮助读者从宏观上对DPDK有一个基本认识。

第1章

背景概述

1.1 产业背景

长期以来，电信业依托各类功能专用的网络设备和庞大的支撑系统组建通信网络，形成了相对封闭的运营环境。这种刚性的运营体系在互联网业务飞速发展的今天，遇到了前所未有的挑战：功能固化的运营设施与业务的灵活适配、稳定的通信标准与新业务的快速部署、粗粒度的网络资源调整与用户/业务资源的按需分配能力等几方面的矛盾日益突出。

互联网的本质特征是以应用为中心和以用户体验为中心，一个架构灵活、富有弹性的网络是连接应用与用户的重要桥梁。随着云计算/大数据应用的不断深化以及网络界对网络重构技术的长期思考，SDN/NFV 这种变革性技术应运而生，为解决上述挑战性问题提供了历史性机遇。

尤其是 ETSI NFV 技术，它使得专有的网络功能能够运行在通用 x86 架构硬件上的虚拟化网络功能（Virtual Network Function，VNF）中，为电信运营商和互联网服务商提供了一种灵活的业务部署手段和高效的组网方案，可以支持固移网络和 IDC 中 NAT（Network Address Translation，网络地址转换）、DPI（Deep Packet Inspection，深度分组检测）、EPC（Evolved Packet Core，演进分组核心网）、防火墙（Firewall）等各类业务功能的广域灵活部署与网元弹性扩展。网元设备未来可能会被“统一的标准化硬件平台+专用软件”所取代，统一标准化的硬件设备有利于降低设备成本与实现资源共享，从而简化网络的部署成本，实现资源的灵活调度。

不同于典型的数据中心业务和企业网业务，电信广域网业务要求网元（如 BNG、DPI 等）具有高吞吐、低时延、海量流表支持和用户级 QoS 控制的特点。大量实践表明，通用 x86 服务器作为 NFV 基础设施用于高速转发业务时，面临着严重的转发性能瓶颈，需要针对性地从硬件架构、系统 I/O、操作系统、虚拟化层、组网与流量调度和 VNF 功能等层面进行性能优化，才能达到各类 NFV 网络业务的高性能转发要求。

根据 ETSI 的 NFV 参考架构，现实中的 NFV 应用系统一般由 NFV 基础设施和 VNF 两类系统服务商提供。因此，相应的 NFV 网络端到端性能测试，也应划分为底层的 NFV 基础设施性能与上层的 VNF 性能两类，以明确各自的性能瓶颈，并避免性能调优工作的相互干扰。

同样，NFV 基础设施又可进一步划分为物理主机（包括操作系统）、虚拟化层和网络 3 个部分，下面简要分析与三者相关的性能问题。

1.1.1 x86 架构性能分析

首先从物理主机（包括操作系统）的性能问题进行分析。大量实践表明：处理器/内存规格、软件代码、操作系统和 I/O 设备的不同是导致物理主机间性能差异的主要原因。

（1）在物理硬件中，处理器架构、CPU 核的数量、时钟速率、处理器内部缓冲尺寸、内存通道、内存时延、外设总线带宽和处理器间总线带宽以及 CPU 指令集等因素，对应用软件（VNF）有较大影响。

（2）多核处理器环境下的软件设计水平也会导致应用软件性能的差异。例如，基于流水线、多线程和防死锁的设计，对底层硬件架构的准确感知，根据 CPU/核的负荷状态分配线程，执行内存就近访问等操作，均可极大提升通信进程的处理性能。

（3）操作系统作为应用软件与物理硬件间的桥梁，通过对物理资源进行合理抽象，不仅提供了多核实时运行环境、内存管理机制、独立内存访问（不同进程间）、文件系统等基本功能，还提供了诸如巨页表管理等高级功能。更大的页表可以减少从逻辑内存页面到物理内存的转换时间、增加缓存命中率，甚至为了获得更快的访问速度，这些地址转译映射可以保存在处理器的内部一个称为 TLB（Translation Lookaside Buffer）的缓存中。但是，获得巨页表和 TLB 支持的前提，是需要对处理器和操作系统（Linux 或者 Windows）进行正确的配置。

此外，将不同的应用软件线程安排在特定处理器核上运行，这种调优方法可以大幅提升软件处理性能。因此，操作系统的高级功能还需要在多个 CPU 核之间提供任务（Task）隔离操作和算法。

在某些场合，应用软件甚至还可能旁路掉部分操作系统的特性，开发出一些实时特性或者特殊的操作库函数。例如，避免操作系统加锁的独立内存结构、基于内部内存流水线的线程间通信机制、旁路操作系统内核协议栈及提供特殊内存读写的 I/O 加速技术、有助于内存就近访问的特殊库函数等。

最后，网卡等外设的驱动程序如何与操作系统连接，并将数据收入内存的过程，对 I/O 性能有重大影响。现代处理器均支持 DMA（Direct Memory Access，直接内存访问）功能，保证外设直接访问系统内存而无需 CPU 参与底层报文的收发过程；但是，在普通网卡驱动程序中，DMA 对每个报文处理完毕后依然会产生一个中断送达 CPU，高速网卡中频繁的硬中断将导致严重的 CPU 阻塞。目前大多数优化技术采用关闭中断的轮询方式，甚至允许 I/O 报文直接访问 CPU 内部缓存。

比较有趣的是，尽管强调正确配置操作系统的重要性，但是由于 I/O 和内存访问的性能对整个 NFV 性能有决定性影响，因此目前的 NFV 性能优化技术往往都是旁路掉操作系统及其相关的 I/O 机制和中断机制。

接下来再分析虚拟化层的相关性能问题。

相比裸金属物理主机，虚拟化层最大的区别就是引入了虚拟化管理程序（Hypervisor）。作为一个软件层，Hypervisor 创建了一个中间抽象层用以屏蔽物理资源（CPU、内存、磁盘和 I/O 设备等）的差异，并将物理硬件分片为虚拟硬件资源，在这些虚拟的分片硬件上可以

运行多个操作系统。Hypervisor 具体功能包括以下几个方面。

- 屏蔽硬件资源并对来宾操作系统（Guest OS）提供统一的虚拟硬件。
- 管理中断：将硬件中断映射至对应的来宾操作系统，反之亦然。
- 转译虚拟内存地址至物理内存地址：Hypervisor 负责将来宾操作系统所使用的来宾物理内存页转译为宿主物理内存地址页，包括 I/O 内存地址块（如 DMA 地址操作）。
- 为来宾操作系统提供一个严格的 CPU 指令集，以保证来宾操作系统运行在一个稳定的仿真硬件平台中。

在环境配置不当时，Hypervisor 必然会引入一定的性能开销，例如成倍的内存页面转译、中断重映射、指令集转译等操作。随着 Hypervisor 技术的演进，这些开销目前已经大为减少，所采用的新技术包括以下几个方面。

（1）采用两级地址转译服务（硬件支持的虚拟化技术），CPU 可以旁路 Hypervisor 而直接访问虚拟内存，并支持包括巨页表在内的地址转译，所以虚拟机内可以透明使用巨页表。

（2）针对 I/O 的 IOMMU（Input Output Memory Management Unit，输入输出内存管理单元）或转译服务：针对 I/O 的两级内存地址提供转译服务和中断重映射、I/O 巨页表转译，从而使 I/O 设备可以旁路 Hypervisor 而直接访问物理内存和 I/O TLB 处理器缓存。

（3）为处理器的一级缓存和 TLB 缓存新增部分新字段以避免缓存刷新，允许不同的虚拟机共享相同的缓存或 TLB 缓存而不会相互影响。

（4）I/O 设备复用（如 SR-IOV）技术则将单一 PCIe 设备分片为多个 PCI 设备，这些分片设备在 PCI-SIG（Peripheral Component Interconnect Special Interest Group）中被称为虚拟化功能（Virtual Function，VF），拥有独立的配置空间、中断、I/O 内存空间等。每个 VF 可以被直接分配给一个虚拟机使用，以避免 Hypervisor 带来的软件上下文切换。

（5）在虚拟机中，对报文收/发队列中的数据进行报文分类，允许一台虚拟机直接管理 I/O 中断，以避免 Hypervisor 中虚拟交换机的每个报文收发产生的单个中断。

上述这些所谓硬件支持（在处理器和网卡中）的虚拟化功能，极大地提升了虚拟化层的处理性能。而且，与物理主机环境类似，虚拟化层同样需要考虑虚拟机与物理主机中的地址映射问题、线程在多核间的分配问题和资源亲和性问题。

1.1.2　NFV 中的网络转发性能分析

在 SDN/NFV 技术的推动下，各类具有高性能转发要求的会话控制类业务功能，如 vBNG、vPGW（40 Gbit/s 以上）、vIMS 等虚拟化业务也逐步加入 NFV 的应用行列。下面通过分析 NFV 基础设施的流量转发性能瓶颈，说明 DPDK 提出的技术背景。

NFV 中的网络业务 App 运行于服务器的虚拟化环境中，单个 App 业务流量的收发要经过虚拟化层、服务器 I/O 通道、内核协议栈等多个处理流程，而多个 App 业务之间又可以用复杂的物理或虚拟网络相连接，称为业务链（Service Chaining）技术。因此，NFV 系统的整体性能取决于单服务器转发性能与业务链转发性能（端到端转发性能）两个方面。

由于采用软件转发和交换技术，单服务器内部转发性能是 NFV 系统的主要瓶颈。如图 1-1 所示，业务 App（VNF）流量的收发 I/O 通道依次包括物理网卡、虚拟交换机、虚拟网卡几个环节（图 1-1 左半部分）。从软件结构上看，报文的收发需要经过物理网卡驱程、宿主机内核网络协议栈、内核态虚拟交换层、虚拟机（VM）网卡驱程、虚拟机内核态网络

协议栈、虚拟机用户态 App 等多个转发通道，存在着海量系统中断、内核上下文切换、内存复制、虚拟化封装/解封等大量 CPU 费时操作过程。

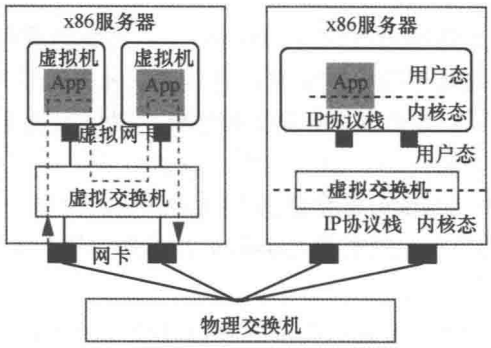

图 1-1　典型的 NFV 应用部署示意

（1）网卡硬件中断

目前大量流行的 PCI/PCIe 网卡在收到报文后，一般采用 DMA 方式直接写入内存并产生 CPU 硬件中断，当网络流量激增时，CPU 大部分时间阻塞于中断响应。在多核系统中，可能存在多块网卡绑定同一个 CPU 核的情况，使其占用率达到 100%。因此，网卡性能改进一般采用减少或关闭中断、多核 CPU 负载均衡等优化措施。

（2）内核网络协议栈

Linux 或 FreeBSD 系统中，用户态程序调用系统套接字进行数据收发时，会使用内核网络协议栈。这将产生两方面的性能问题：一是系统调用导致的内核上下文切换会频繁占用 CPU 周期；二是协议栈与用户进程间的报文复制是一种费时的操作。NFV 系统中，业务 App 报文从物理网卡到业务 App 需要完成收发操作各 1 次，至少经过多次上下文切换（宿主机 2 次以及 VM 内 2 次）和多次报文复制。将网络协议栈移植到用户态是一种可行思路，但这种方法可能违反 GNU 协议。因此，弃用网络协议栈以换取转发性能是唯一可行的办法，但需要付出大量修改业务 App 代码的代价。

（3）虚拟化层的封装效率

业务 App 中存在 2 类封装：服务器内部的 I/O 封装和网络层对数据报文的虚拟化封装。前者是由于 NFV 的业务 App 运行于 VM 中，流量需要经历多次封装/解封装过程：宿主机 Hypervisor 对 VM 的 I/O 封装（如 QEMU）、虚拟交换机对端口的封装（如 OVS 的 tap）、云管理平台对虚拟网络端口的封装；后者是为实现 NFV 用户隔离而在流量中添加的用户标识，如 VLAN、VxLAN 等。这 2 类封装/解封均要消耗 CPU 周期，从而降低了 NFV 系统的转发效率。

（4）业务链网络转发效率

NFV 的业务链存在星型和串行 2 种组网方式，如图 1-2 所示。星型连接依赖物理网络设备的硬件转发能力，整体转发性能较优，但当 App（运行于 x86 服务器）的数量较大时，会消耗大量昂贵的网络设备端口。因此，在业务链组网范围不大时，如在广域网的业务 PoP 点，为简化组网和节约端口，可以考虑采用串行连接。在串行组网方式下，不合适的负载均衡算法会造成流量在不同进程组的上下行链路之间反复穿越，严重降低业务链网络的带宽利用率。