<https://cloud.tencent.com/developer/article/1187579>

DPDK在专注数据面报文处理的同时，一直紧跟着网络发展的脉搏以开放的姿态融合不断涌现的各种新的网络设备。从最初的普通网卡，到集成虚拟化和交换功能的高级网卡，再到各种网络SoC（片上系统）设备，到现在最热的基于FPGA的Smart NIC，DPDK一直走在软件定义的网络技术发展的最前沿。近年来，数据中心异构化的趋势出现，基于云的数据中心如何使用加速器来进行存储，网络以及人工智能的加速，成为炙手可热的话题，在刚结束的APNET’18研讨会上，华为与腾讯都分享了技术方向与实践演进过程，基于Linux Foundation的开源项目，对这种架构的支持，在软件的持续性与高质量保证上至关重要。



本文首先分析FPGA在[网络加速](https://cloud.tencent.com/product/dsa?from=10680)中的优势，阐述了FPGA部署的软件基础架构的支撑，然后介绍英特尔FPGA加速开发栈，最后详细说明DPDK 18.05版本中发布的新特性，这是第一款支持FPGA加速的通用网络软件框架。

FPGA为网络业务提供卓越的性能加速

毋庸置疑，FPGA(Field Programmable Gate Array)已经是数据中心计算和网络加速的宠儿，可以灵活地通过软件更改、配置器件内部连接结构和逻辑单元，完成既定设计功能的数字集成电路芯片不仅可以达到近似ASIC的性能，而且可在不到一秒的时间进行再编程，其数据路径可精确匹配灵活多变的适应用户工作负载，提升数据中心的性能、功效和利用率，长期看，可以有效降低数据中心的总体运维成本， 减少基于专用ASIC加速的重复投资。微软是这方面的先行者，发表了大量论文与学界、工业界分享进展与挑战 。

FPGA在网络加速中的优势概括起来有三点：灵活性、并行性和低延迟。相对于研发成本高，研发周期长的ASIC，可配置的FPGA无疑是非常灵活的。不同于冯·诺依曼结构，FPGA 同时拥有流水线并行和数据并行，例如处理一个网络包，一般要经过解析、分类、查表和修改这几个基本步骤，复杂点的还需要流量管理、限速和报文复制等功能。FPGA可以通过搭建针对每一级处理的流水线，实现报文处理指令和报文数据的并行进而加速网络。低延迟得益于FPGA的精确可控报文处理步骤，在实时性要求高的领域比如金融有着非常大的市场。

不过FPGA也不是万能的，相比于软件开发，FPGA的开发周期要长很多，一个FPGA版本的开发需要经过编译、分析综合、布局布线和时序分析等多个步骤，一般都是以小时为单位计算。至于调试，也没有GDB的单步调试来的方便，FPGA的在线调试全靠芯片内置的逻辑分析仪抓波形，通过分析“天书”一样的波形图来解决。另外，受限于片内的资源，FPGA在实践中主要聚焦在网络底层的报文处理。

就FPGA的开发而言，从功能上可以分为可固化的接口部分和灵活多变的用户业务逻辑部分，接口部分的开发如PCIe、DDR、以太网MAC和PHY与硬件紧密相连且一旦单板做好改动较少，而用户业务逻辑会随着新功能的加入而不断演进和变化，这才是体现FPGA的灵活性的地方。

从上面的分析我们可以看到，FPGA的部署和应用是需要一些基础架构层面的支撑，这些支撑有软件方面的也有FPGA方面的，软件方面又分为公共软件架构部分和FPGA芯片特定的软件部分。在接下来的章节中，我们将对此做详细的描述。

英特尔FPGA加速开发栈

这部分关注FPGA和芯片特定软件的基础架构支撑部分，图1描述了系统架构与开发层次，开发者可以依据项目需要，安排软件开发人员在适合的层次上轻松地利用英特尔FPGA。Intel发布了OPAE软件, 提供可从虚拟机和容器调用的线程安全应用编程接口 (API)，能高效隔离软件与FPGA开发者的相互依赖。利用基于DPDK的软件接口API，可以无缝支持当前与后续的FPGA设备，有效保证了软件的移植性与独立性。RTL开发者也能清晰依赖软件接口来进行基于模块的独立开发。

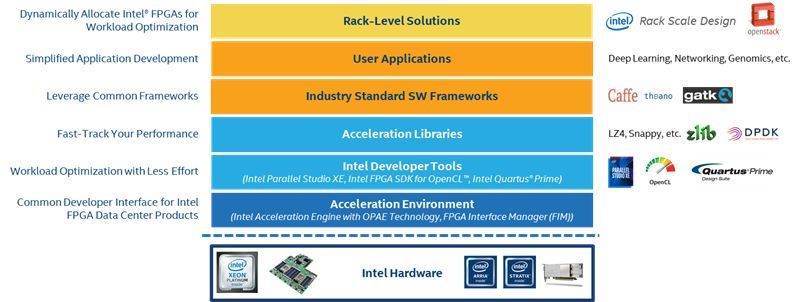


图1英特尔FPGA加速堆栈

从应用模式上来看，如图2所示，整个FPGA分为固化的蓝色部分(FPGA Interface Manager)，和灵活可变的绿色部分(Accelerated Function Unit)。蓝色部分主要包含FPGA外围接口和FPGA配置管理，绿色部分代表的是用户自定义逻辑，一片FPGA可以根据资源大小提供一个或多个AFU（加速功能处理单元）。这种应用模式可为设计工程师带来诸多优势：

* 节省开发人员的时间，使其专注于提高解决方案的价值
* 支持跨多个英特尔 FPGA 外形产品重用代码
* 构建全球首个用于英特尔 FPGA 数据中心产品的常见开发人员界面
* 灵活的FPGA池化和资源共享成为可能

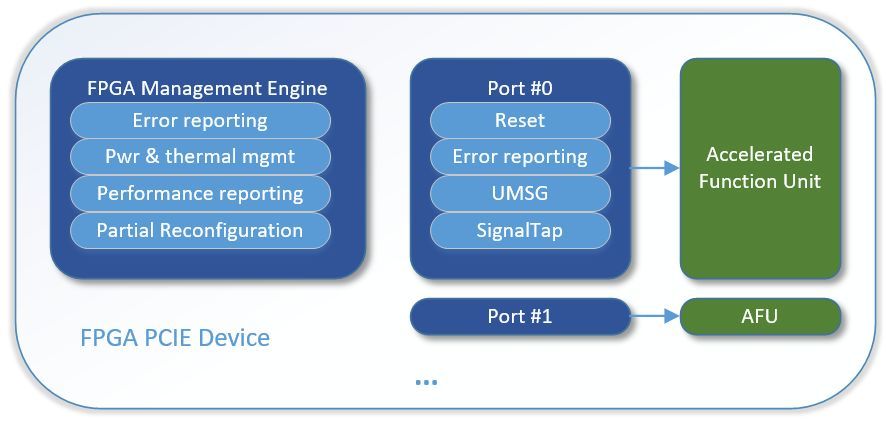


图2 FPGA Interface Manager

开放可编程加速引擎 (OPAE) 技术是一个软件编程层，提供了一个适用于各代 FPGA 产品和平台的 API。它旨在最大限度降低软件开销和延迟，同时为硬件特定的 FPGA 资源详细信息提供一个抽象层。为了培养开放的生态系统并鼓励使用 FPGA 加速功能处理数据中心工作负载，英特尔向行业和开发人员社区开放了这一技术。

* 开放可编程加速引擎技术的特性：
* 轻量级的用户空间库 (libfpga)
* 许可：FPGA API (BSD)、FPGA 驱动程序 (GPLv2)
* FPGA 驱动程序向上游移动到 Linux\* 内核
* 支持容器，虚拟机和裸机平台
* 包括指南、命令行实用程序和示例代码

DPDK的 FPGA加速框架

针对FPGA的网络加速应用，DPDK 18.05版本正式发布了第一款支持FPGA加速的通用软件框架。这套框架解决了FPGA应用时遇到的3个最重要需求：

* 在DPDK里支持FPGA管理的设备类型
* 如何将DPDK的高性能驱动与AFU（加速功能单元）关联
* FPGA加速设备的引入要能最大限度的兼容开源API,比如DPDK API

对于FPGA的管理，DPDK 18.02的发布版本中新提供的rawdev类型结构，可以提供很好的适配。rawdev的优势主要体现在两点：业务无关性和丰富的操作接口。FPGA作为一种灵活可配置的芯片，它可以通过下载不同的Bitstream实例化不同类型的加速设备，如网络加速设备和加解密加速设备，因此FPGA管理必须由一种业务不感知的类型结构来操作，rawdev正好切合了这一点。此外，在rawdev内部提供了大量的操作接口如固件下载、属性获得、设备管理等可以用来作为FPGA的管理接口。

对于DPDK驱动与AFU的关联，我们提出了FPGA-BUS这一新的通用总线。在FPGA基础架构层面的支撑下，FPGA作为一个PCIe设备不再与业务关联，取而代之的是FPGA内部的AFU与用户驱动的绑定。在实现时，我们的出发点是尽量的复用现有DPDK PCI BUS Scan流程，最好不修改。事实上，通过引入FPGA-BUS，我们有效地做到了这一点。用户只需要编写基于AFU的软件驱动而不是PCI驱动，每个AFU都有自己的设备ID，AFU驱动通过AFU ID与AFU设备关联。如图3所示，引入FPGA-BUS后，整个驱动的加载以两次扫描的形式清晰的呈现给用户：PCI BUS 扫描和 FPGA BUS 扫描。

* PCI BUS 扫描: FPGA 作为一个PCIe设备被扫描并加载其rte\_rawdev驱动，这是标准的DPDK PCI BUS 扫描
* FPGA BUS 扫描: rawdev根据用户配置载入不同的AFU并Hot Plug触发FPGA BUS 的扫描，从而加载AFU 驱动

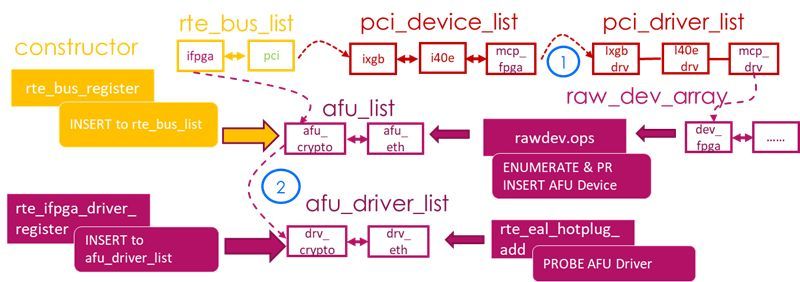


图3 两次BUS 扫描和AFU驱动加载

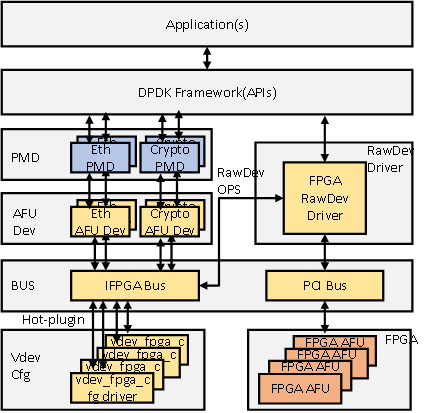


图4 DPDK FPGA加速框架示意图

整个加速框架的详细架构如图4所示，我们引入vdev来进行AFU配置参数的载入，FPGA与其rawdev关联，AFU驱动与AFU设备关联。由于rawdev屏蔽了硬件细节和FPGA BUS的引入，DPDK 应用感知不到它所调用的是FPGA加速设备还是普通的NIC加速设备。

英特尔FPGA加速堆栈与DPDK FPGA加速框架的集成

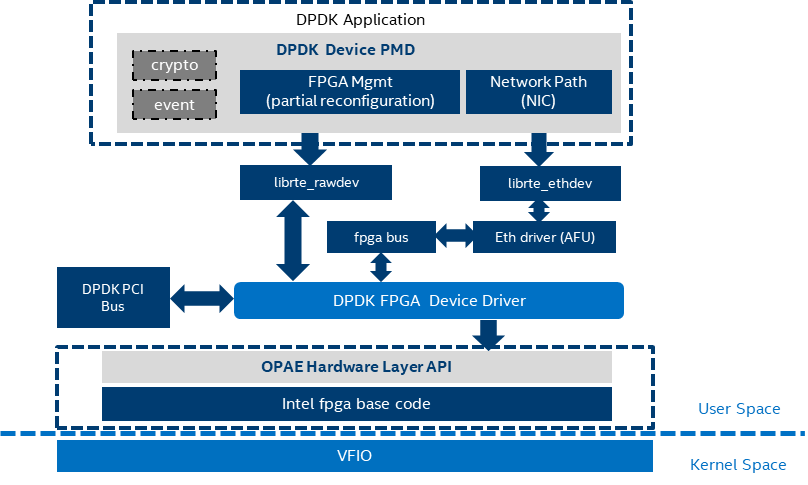


图5 英特尔FPGA加速堆栈与DPDK FPGA加速框架的集成

如图5所示，英特尔FPGA加速堆栈的软件部分OPAE以Share Code的形式提供给DPDK，DPDK 的rawdev结构通过OPAE管理和配置FPGA，FPGA-BUS提供对AFU的管理和驱动加载。基于这套架构，DPDK提供了软件框架可以更轻松地开发和部署英特尔 FPGA设备，优化数据中心的工作负载。在报文处理领域，如果把DPDK比作“倚天剑”的话，那么FPGA就是“屠龙刀”，这两者强强联手威力无穷，持有这两个武器的开发者，在数据中心网络加速中可以说是无人能敌。