**硬件卸载**

**调研报告**

其中最为知名的一个当属Kernel Bypass技术，即跳过内核的网络层并把所有的网络包处理交由用户空间执行。Kernel bypass同时要求在用户空间管理网卡，或者说，网卡要依赖用户空间中的驱动来发挥功能。

将网卡的完全控制权交给用户空间程序的好处是，我们减少了内核所带来的障碍，比如上下文切换、网络层处理、中断等。这足以使网络数据传输达到10Gbps甚至更高。kernel bypass与如批量包处理等其他方案和性能调整方法（如NUMA感知和CPU隔离等），整合除了高性能用户空间网络的基础结构。这种新的数据包处理方法的典型代表应当是英特尔的DPDK，同时还有其他为人熟知的VPP，PF\_Ring和Snabb。

而用户空间网络的不足则在于：（1）用户空间程序跳过了操作系统内核，就意味着跳过了操作系统提供的硬件资源抽象而直接管理硬件本身。用户空间驱动程序虽然可以保证正常运作，但一般地相比内核更缺乏测试与兼容性。（2）用户空间程序同时跳过了内核提供的网络管理函数，这意味着用户空间程序要重新实现那些已经被内核提供过的功能。（3）程序以沙箱模式运行，而这限制了它与操作系统其他部分的集成于交互。（4）内核为网络提供充分的安全层，而这在用户空间程序中并不存在。

与用户空间网络截然相反的是，XDP（eXpress Data Path）将诸如过滤器，映射器，路由的全部用户空间网络程序转移到了内核的领域里。XDP允许我们的网络程序在网络包到达网卡而进入内核网络层之前立即执行，这显著提高了网络包处理速度。而为了让用户程序在内核领域内执行，BPF（Berkeley Packet Filtering）进行了有效的实现。

BPF实质上是一个特定功能的虚拟CPU模型，这个虚拟机是为包过滤处理而特别设计的。BPF虚拟CPU及其字节码是1992年末在论文“The BSD Packet Filter: A New Architecture for User-level Packet Capture”中由Steve McCane和Van Jacobson首先提出,BPF虚拟机定义了BPF程序执行的环境：除字节码之外，还定义了一个基于数据包的内存模型,其中“加载”指令在处理数据包上隐式执行，A与X（Accumulator and Index register）两个寄存器，临时存储器存储和隐式程序计数器。在2016年，东京netdev 1.2大会上，Jakub Kicinski与Nic Viljoen提出了将eBPF/XDP程序硬件卸载到NETRONOME的数据流处理器智能网卡上的架构方法。

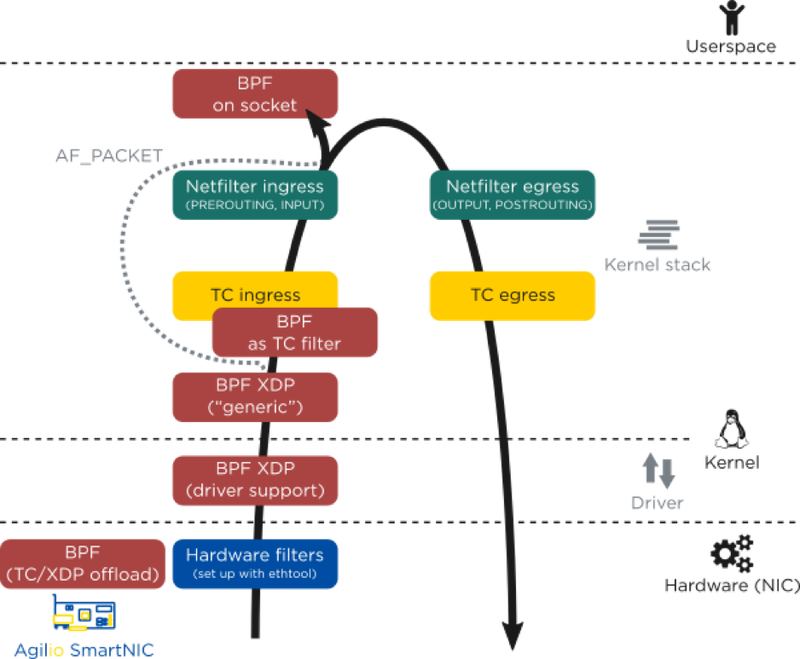
在以NPU为基础的SmartNIC出现之前，由于传统NIC缺乏广泛的硬件卸载适配性，并且传统x86通用CPU已经能较好地实现硬件卸载，同时实现通用卸载要适配多种不同特定硬件架构的复杂性，少有成功的eBPF向NIC的硬件卸载。而随着通用CPU难以胜任目前的网络负载规模，并且RISC工作者在NPU为基础的SmartNIC上的工作日趋成熟，向SmartNIC上硬件卸载eBPF程序正当其时。

**可行性报告**

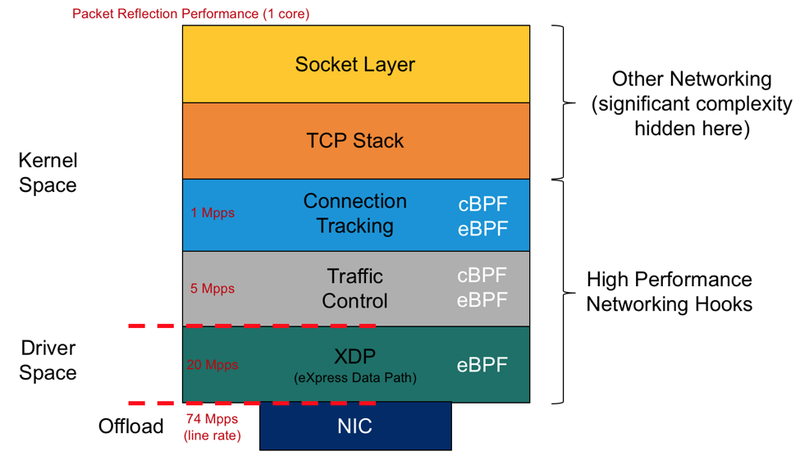
BPF获得硬件卸载的支持之前，这些架构通过在整个 Linux 的网络通路上添加hook，来获得更早处理网络数据的机会。

hook 应当解释为一个“嵌入”的结构：当数据到达数据路径的某一个节点时，挂在这个“嵌入”结构上的架构程序立刻开始执行，比如处理该数据包是否应该丢掉——而非等到数据包创建了 skb 之后再进行决定。这种嵌入架构允许我们提前做出决策，规避毫无意义的硬件、内核层数据传输和复制。

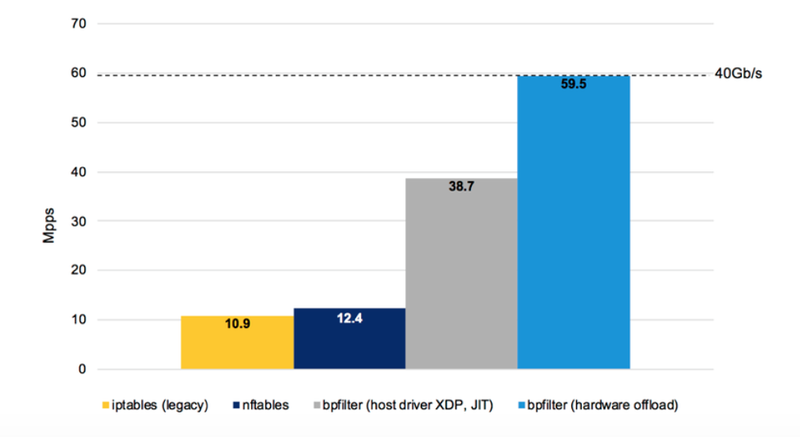
eBPF 程序的特性，支持它可以和内核的诸多架构配合，在网络数据通路的诸多节点上嵌入其程序。下图展示了 eBPF 程序可以嵌入的内核架构（同时包括了网卡驱动和硬件卸载）。



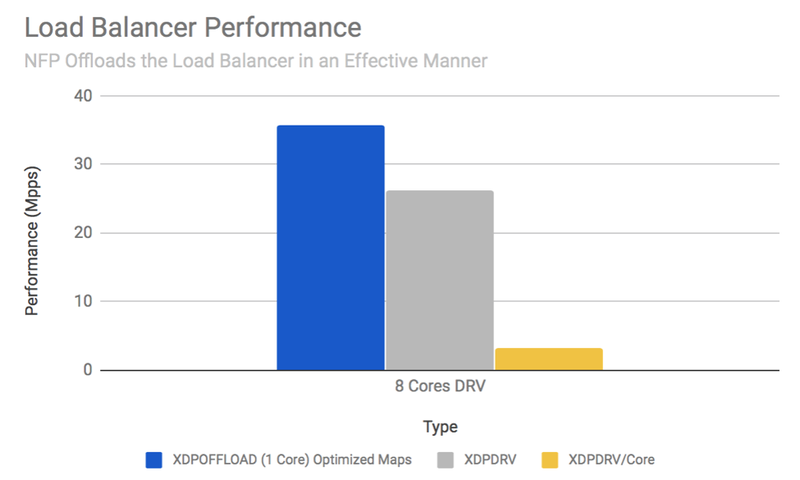
近年来高热发展的 Linux 网络数据处理架构中，TC 和 XDP 都选择将其 hooks 嵌入到网络数据通路中尽可能低的部分。下图展示的是所有使用 eBPF 的高速网络处理架构在数据通路的位置及单核包处理表现。



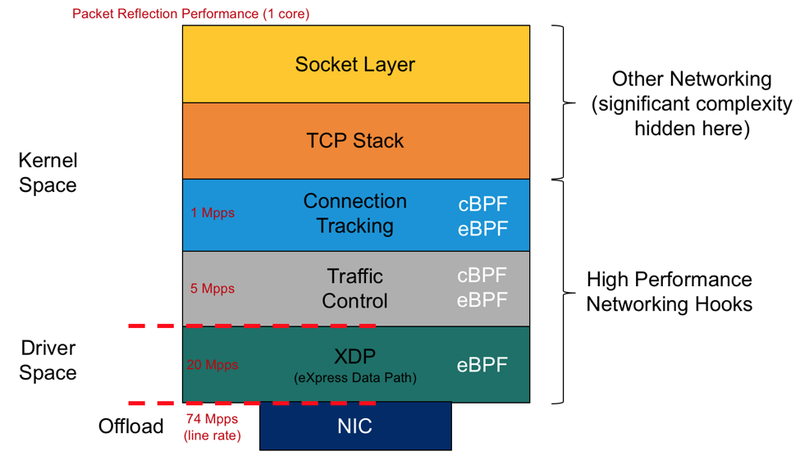
可以看到，层次越低，包处理的速度越快，而以硬件卸载方式单线运算速率是 XDP 的进四倍，具有很大的性能优势。

另一eBPF 硬件卸载的发展结果是 bpfilter，bpfilter是一个同时兼顾加入 eBPF 的新特性与兼容固有防火墙与 ip 协议的高性能网络过滤内核模块。下图是展示出 bpfilter 硬件卸载到 SmartNIC 上处理数据，与使用八核处理器以及使用旧有的传统 iptables 和较新的 nftables的数据处理速度的性能差异，使用bpfilter的数据处理速度明显优于其他方案。

基于数据流处理器的网络处理硬件卸载的可以提高性能的原因在于：它规避了如x86等传统架构在 PCIe 带宽限制上不可避免的障碍。下图是使用 XDP 执行负载均衡时，使用 Agilio SmartNIC 硬件卸载与网卡驱动层XDP，及使用 Intel Xeon CPU E5-2630 的性能对比，前者的数据包处理表现近乎是后者单核的12倍。



同时，低延迟性也是选择硬件卸载 eBPF 的关键理由。由于 eBPF 程序直接在网卡上运行，数据包不必在跨越 PCIe 带宽造成的障碍，进而达到改善负载平衡和维护 DDoS 网络安全。下图展示了 XDP 在硬件卸载和网卡驱动层两个方式下的延迟对比，硬件卸载的延迟十分稳定。



综合上述内容得出结论：eBPF 程序通过已有的官方工具，可以方便地编程到高度支持硬件卸载的 Agilio SmartNICs 上，并有效规避传统冯诺依曼架构中数据传输和计算的瓶颈，利用数据流处理器的高性能和高度并行性，获得高性能网络处理所需要的更高速的包处理和更低的延迟。