PVPP: A Programmable Vector Packet Processor
Sean Choi, Xiang Long, Muhammad Shahbaz, Skip Booth, Andy Keep, John Marshall, Changhoon Kim

1. Problem Statement
- Programmable packet forwarding abstractions used for software switches, such as the match-action abstraction, are not expressive enough to fully exploit the underlying CPU architecture.
- Many switch targets still expose just enough interfaces for customizing the match-action pipeline, but not a lot more.
- What if the switch targets expose more complex interfaces for interacting with the underlying architecture for finer tuning of compilers for increased performance?

2. Approach

VPP as the Programmable Software Switch Target
- Vector Packet Processing (VPP) technology’s unique node graph packet processing model allows a various number of packets, combined in a single packet vector, to be processed arbitrarily at each node with separate and isolated instructions.
- VPP exposes low-level interfaces for directly interacting with the CPU and memory.
- Programmable Vector Packet Processor (PVPP), an extension of VPP, is a programmable software switch with a programmable node graph packet processing abstraction.

PVPP Architecture Overview
- PVPP is programmed via P4, a domain specific language specially designed to easily describe data plane behavior.
- PVPP’s P4 highly configurable compiler generates customized VPP nodes.
  - Some example of configurable parameters.
    - Number of nodes for the given P4 file.
    - Number of packets per vector or per iteration.
  - P4 Headers map directly to PVPP’s C structs for increased readability and performance.

3. Experimental Setup

Experimental Platform Topology

Experimental Server Specifications
CPU: Intel Xeon E5-2640 v3 2.6GHz
Memory: 32GB RDIMM, 2133 MT/s, Dual Rank
Hard Disk: 1TB 7.2K RPM NLSAS 6Gbps
NICs: Intel X710 DP/QP DA SFP+ Cards

4. Results

L2-L3 Benchmark Application
- The experimental results on L2-L3 benchmark application show that optimized PVPP has comparable throughput performance with VPP and other P4 to software switch implementation.

Compiler Optimization Results
(64 byte packets over a single 10G port)

Throughput Comparison with PISCES
(over all six 10G ports)