Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Bachelor's Thesis: Performance-Analysis of VPP

Intermediate Talk

Presenter: Peter Okelmann— *Advisors:* Paul Emmerich, Dominik Scholz— *Supervisor:* Prof. Dr.-Ing. Georg Carle

VPP: a fast software router

VPP (Vector Packet Processing) is a user-space software router. This approach combines many advantages:

- deployable to usual architechtures
- fast user-space network interface drivers
- can run in virtualized containers

"It is the open source version of Cisco's Vector Packet Processing (VPP) technology."[1] Now it is beeing developed by FD.io ("The Fast Data Project") which belongs to the Linux Foundation.
Feature Highlights:

- vecorized processing of packets in badges
- utilizes high-speed dpdk drivers
- modular and extendable packet-processing graph
- cpu-scalability

Testing Methodology

MoonGen[3] is scripted to generate testing load according to the following testing parameters:

- packet rate
- packet size
- traffic type (generic Ethernet, UDP)
- traffic pattern (inter packet gaps)

Gathered testing results:

- latency histogram
- throughput
- linux perf stats (cache misses...)
- linux perf record (cpu-time spent per symbol)
- internal vpp state information

For tests to return meaningful results, the optimum of throughput to latency is beeing found by a script. This packet rate is then used for further tests. Otherwise the results are inaccurate, because of utilization of the packet buffer. VPP properties to test:

- raw forwarding throughput
- latency: cache and memory impact
- processing graphs utilizing m
- specific processing nodes /

Measurement setup

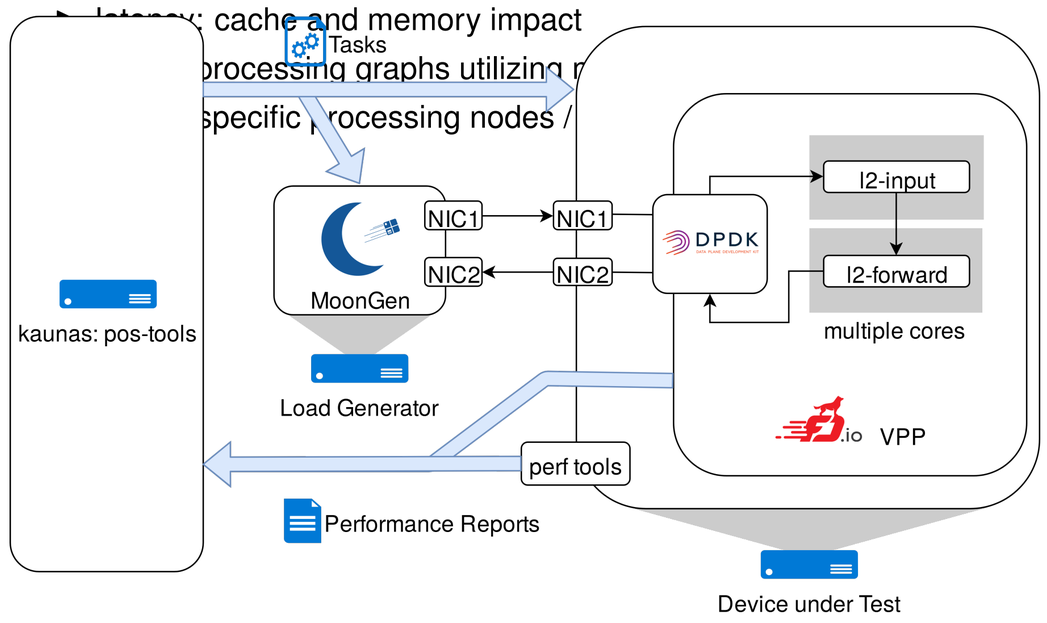
Background

- Important background for this work
- may be some things about related work or important libraries/frameworks used

Measurment Setup:

- How does your setup look like (maybe a figure)?
- What are the relevant questions you try to answer with your measurement?
- What do you measure?
- How do you measure?

Benchmarking Setup



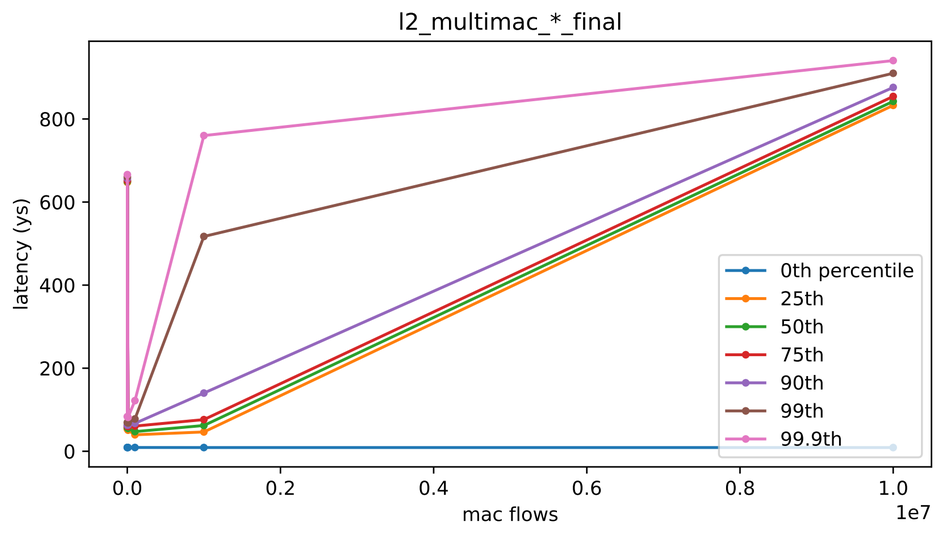
I2 Throughput

The following results are produced by sending non-ip ethernet packets to a single destination. VPP ran on an Intel E5-2640 @ 2.0GHz (cesis) with 10G networking results in the following numbers:

VPP config	max Mpps	stable Mpps	Relative
packet gen speed	14.86	14.86	100%
l2 xconnect	10.4	10.1	68%
l2 bridge: no features	9.35	9.2	62%
l2 bridge: mac-age	8.62	8.6	58%
l2 bridge: mac-learn	8.51	8.3	56%
l2 bridge: mac-learn, mac-age	8.50	8.3	56%

Router	Mpps	Relative
MoonRoute	14.6	100%
FastClick (DPDK 2.2) [3]	10.4	72%
Click (DPDK 2.2) [3]	4.3	29%
Linux 3.7	1.5	10%

Planned Schedule



Short time schedule for the upcoming weeks:

- Official start date: October 15, 2010
- Official end date: February 15, 2011
- Weeks left: 8

Schedule

- Week 1-4: Providing cookies for I8
- Week 5-6: Perform additional measurements
- Week 7: Writing thesis
- Week 8: **Several** corrections passes
- Week 9: Print and hand-in

[1] What is vpp? https://wiki.fd.io/view/VPP/What_is_VPP%3F. Accessed on 2019-01-16.
[2] L. Braun, A. Didebulidze, N. Kammenhuber, and G. Carle. Comparing and Improving Current Packet Capturing Solutions based on Commodity Hardware. In *Internet Measurement Conference 2010 (IMC'10)*, Melbourne, Australia, Nov. 2010.
[3] P. Emmerich, S. Gallenmüller, D. Raumer, F. Wohlfart, and G. Carle. MoonGen: A Scriptable High-Speed Packet Generator. In *Internet Measurement Conference 2015 (IMC'15)*, Tokyo, Japan, Oct. 2015.
[4] P. Emmerich, D. Raumer, F. Wohlfart, and G. Carle. A Study of Network Stack Latency for Game Servers. In *Proceedings of the 13th Annual Workshop on Network and Systems Support for Games*, Dec. 2014.
[5] S. Günther, M. Riemensberger, and W. Utschick. Efficient GF Arithmetic for Linear Network Coding using Hardware SIMD Extensions. In *Proceedings of the International Symposium on Network Coding (NetCod)*, Aalborg, Denmark, June 2014.