The Flow Processing Company

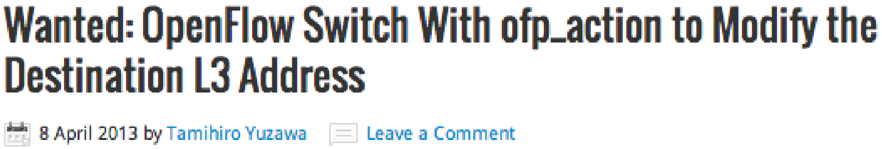
Selective and transparent offload of OpenFlow switches

Rolf Neugebauer

rolf.neugebauer@netronome.com

Problem

- Hardware OpenFlow switches are fast but inflexible:
  - Most implement the required OpenFlow 1.0 features with many optional features missing
  - Recently on packetpusher.net:



Wanted: OpenFlow Switch With ofp_action to Modify the Destination L3 Address

8 April 2013 by Tamihiro Yuzawa Leave a Comment

- OpenFlow is rapidly evolving
  - 1.3/1.4/1.5 are significantly more complex with more features added
- Evolution of SDN in general
  - Overlays with edge switches and a "dumb" fabric
    e.g. Casado et al: "Fabric: A Retrospective on Evolving SDN", HotSDN 2012

➡ Rapid evolution requires the flexibility of a software implementation

- Existing software implementations perform rather poorly

Approach

- Start with a mature software implementation
  - Open vSwitch in our case
- Pair it with a highly programmable network card

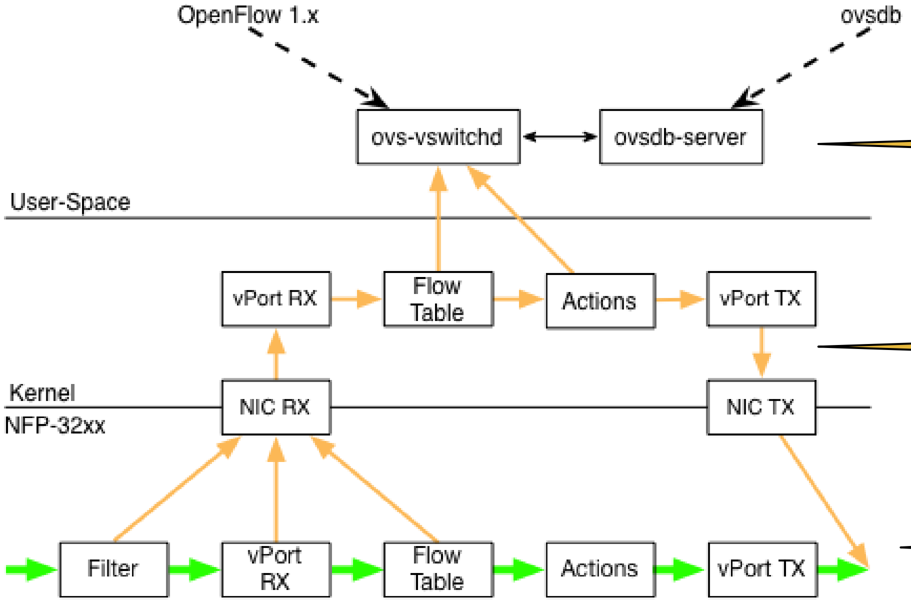
- Selective offload to the network card
  - Implement *some* OpenFlow switch functionality on the network card
  - Other Packets are processed by the software switch implementation as normal

  ➡ Some packets (or features) are handled entirely on card

  ➡High performance PCIe interface essential

- Initially aim for minimum changes to software switch
  - Enables to follow the rapid evolution of the software
  - Tradeoff between accelerated feature set and being able to track changes
  - Enable more offloads over time

Implementation (based on OVS 1.7)



Unmodified userspace code

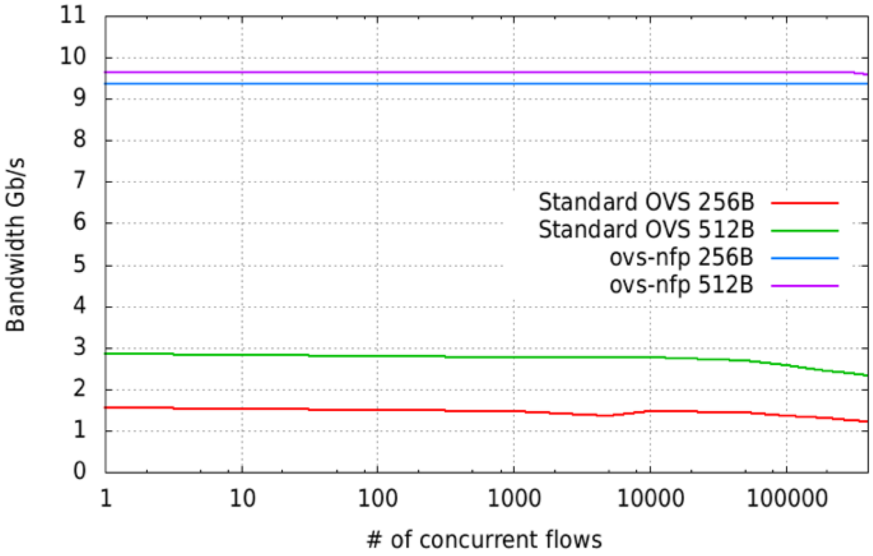
12 hooks in OVS kernel datapath

Partial OVS datapath implementation on the NFP-32xx

- Exact match on full IPv4 OVS flow key, full support for all OVS stats
- All packet modification actions
- GRE tunnel vPorts support
- Implemented in ~1000 lines of C code on the NFP-32xx

Preliminary performance evaluation

Forwarding performance



- L2-in-L3 Gateway configuration
  - GRE decapsulation -> VLAN push
- Best case scenario: Most packets handled on fastpath
- Standard OVS: 650-519Kpps
- ovs-nfp: 3.98Mpps (Line rate)

New connection rate

	Standard OVS	ovs-nfp
vSwitch	32.9K	35.1K
GRE vPort	27.4K	26.8K

- ovs-benchmark rate
- “Worst-case” scenario
- All packets handled by the host
- Host CPU is the bottleneck

System configuration: 6 core Intel Xeon E5-2630 with 16GB RAM. Intel 82599EB 10G NICs, 1.0 GHz NFP-32xx w/ 4GB RAM. Ubuntu 12.04, Linux 3.2.0, OVS 1.7

Summary

- Host based implementation for maximum flexibility
  - Necessary to keep track with rapid evolution
- *Selective* offload to a highly programmable network card
  - Necessary to achieve good performance in the common case
- Offload feature set should be driven by workloads
  - The common case varies between deployments
- Prototype implementation targeted at edge switches in overlays
  - Minimal changes to OVS code base
  - Simple implementation on network card
  - Achieves line-rate forwarding
- Plans to extend to more offloads

Thank you!

Questions?