The Acceleration of OfSoftSwitch

Nicola Bonelli**, Gregorio Procissi**, Davide Sanvito**, Roberto Bifulco**

* Università di Pisa, Pisa (Italy)
† CNIT - Consorzio Nazionale Interuniversitario per le Telecomunicazioni (Italy)
‡ Politecnico di Milano, Milano (Italy)
* NEC Laboratories Europe, Heidelberg (Germany)
Motivations

Software switches are widespread tools for experimenting novel programming paradigms and abstractions

**OfSoftSwitch (OFSS)**

- ✓ very popular tool (150+ GitHub forks)
- ✓ simple and straightforward user-space OpenFlow 1.3 implementation
- ✓ enables fast experimentation
- ✖ not performance-oriented
- ✖ limited to functional experimentation
Contributions

- Acceleration of OfSoftSwitch (aOFSS)
  - 90x performance speedup

- OFSS simplicity preserved
  - new MAT-based forwarding abstractions can be easily implemented
  - Accelerate existing prototypes (we successfully ported OpenState to aOFSS)
OFSS architecture

ofprotocol

- datapath configuration
- communication with controller

ofdatapath

- single process application
- netdev library
- standard Linux AF_PACKET sockets
  - inefficient I/O speed
OFSS acceleration

- Replacement of I/O framework
  - netdev library replaced with pcap library
  - enabled support for PF_RING ZC, netmap, DPDK, PFQ

- ofdatapath code optimizations
  - dynamic memory allocation refactor
  - hash maps refactor
  - zero copy
  - batch processing

- multi-core processing
  - limited modifications to OFSS
  - PFQ framework
PFQ - Packet Family Queue

- Open-source Linux kernel module
- Software-accelerated packet I/O
- In-kernel early stage packet processing
  - Filtering, logging, forwarding, load-balancing, dispatching
  - Programmable via pfq-lang eDSL
- Group abstraction
- Fine-grained packet distribution
  - Application sockets
  - Network interfaces
  - Kernel Network Stack
From OFSS to multi-core OFSS

- PFQ allows to distribute workload across unmodified ofdatapath processes
  - No need to include an additional packet distribution layer into the application
  - Each OFSS instance processes a quota of the traffic according to a hash-based load balancing algorithm
  - ofprotocol presents to the controller a single ofdatapath instance
Performance evaluation

- OFSS vs aOFSS
- OpenFlow pure forwarding

![Diagram showing network setup with SWITCH, TRAFFIC GEN, and CONTROLLER connected by 1G and 10G lines. A graph compares throughput (Gbps) for different packet sizes (64, 128, 256, 512, 1024, 1500 bytes) across Line Rate, OFSS, aOFSS-1 core, aOFSS-2 cores, aOFSS-3 cores, and aOFSS-4 cores.]
OpenState/BEBA switch

- Stateful OpenFlow
  - FSM abstraction
- In-switch forwarding behaviour adaptation
- Stateful stage
  - State table + flow table
- Custom flow definition
From stateless aOFSS to accelerated OpenState

- PFQ allows custom steering logic
  - User-defined consistency via pfq-lang
- OpenState stateful processing based on per-flow state
  - User-defined flow definition

Configure PFQ to steer traffic according to lookup-scope definition
  - each flow with its own state is guaranteed to be processed by the same OFSS instance
  - no need of state synchronization between instances
Performance evaluation (3)

Stateless OpenState on aOFSS

Stateful OpenState on aOFSS
Conclusion

- Acceleration of OFSS
  - aOFSS
- Stateful MAT abstraction acceleration
  - Openstate porting from OFSS to aOFSS
- Open-source contribution
  - aOFSS is available in the BEBA-EU branch of OFSS official repo
  - https://github.com/CPqD/ofsoftswitch13/tree/BEBA-EU
Thanks!
davide.sanvito@polimi.it
Performance evaluation (2)

- Optimization techniques contribution for 1 core