A High-Performance OpenFlow Software Swich

R. Rahimi, M.Veeraraghavan
University of Virginia

Y. Nakajima, H. Takahashi
NTT Labs

Y. Nakajima, S. Okamoto,N. Yamanaka
Keio University

mvee@virginia.edu

Internet2 TechX

Sept. 27, 2016

Miami, FL

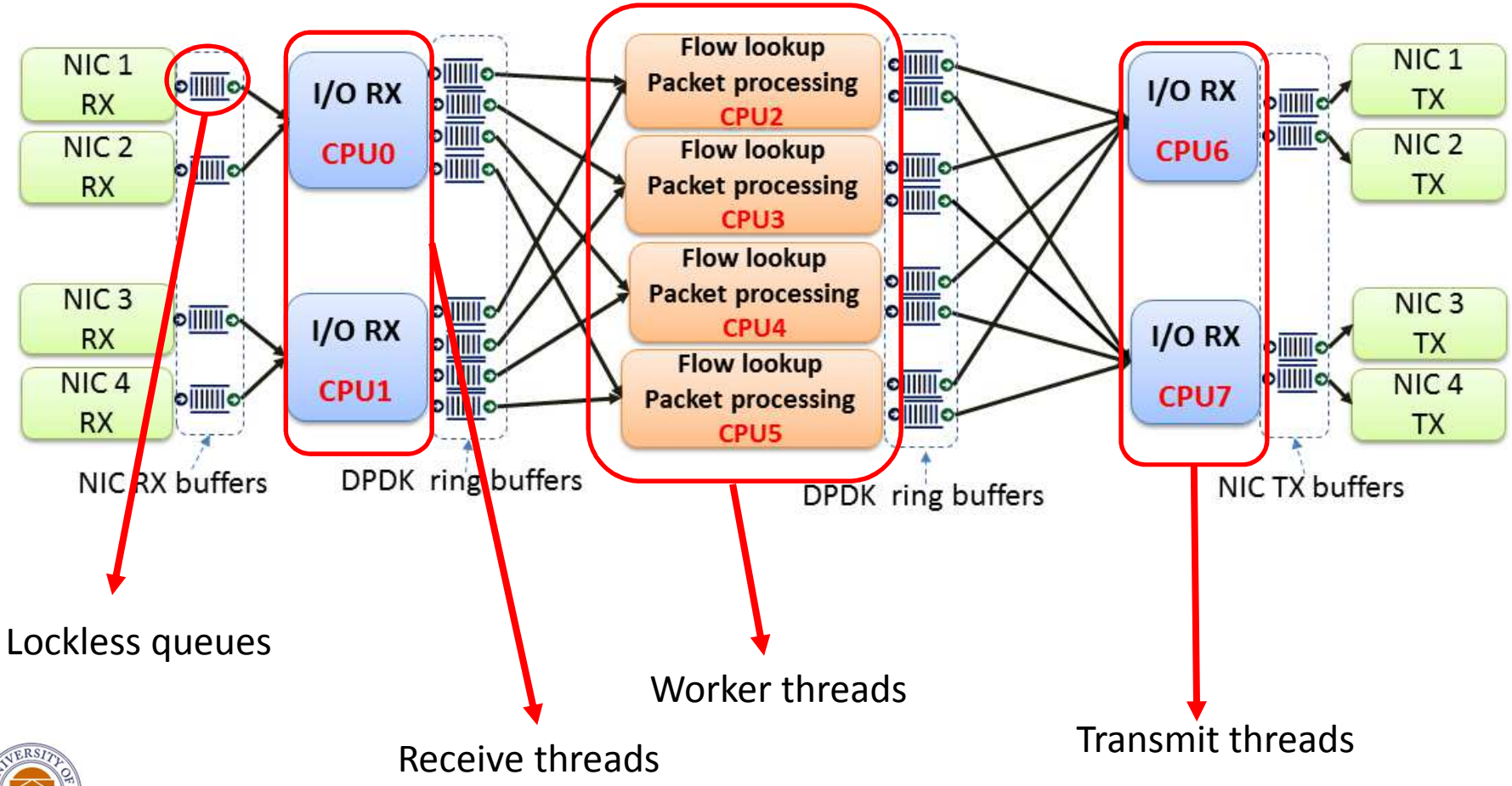
Thanks to NSF for grants ACI-1340910, CNS-1405171, CNS-1531065 and US DOE grant DE-SC0011358, and NICT

Lagopus

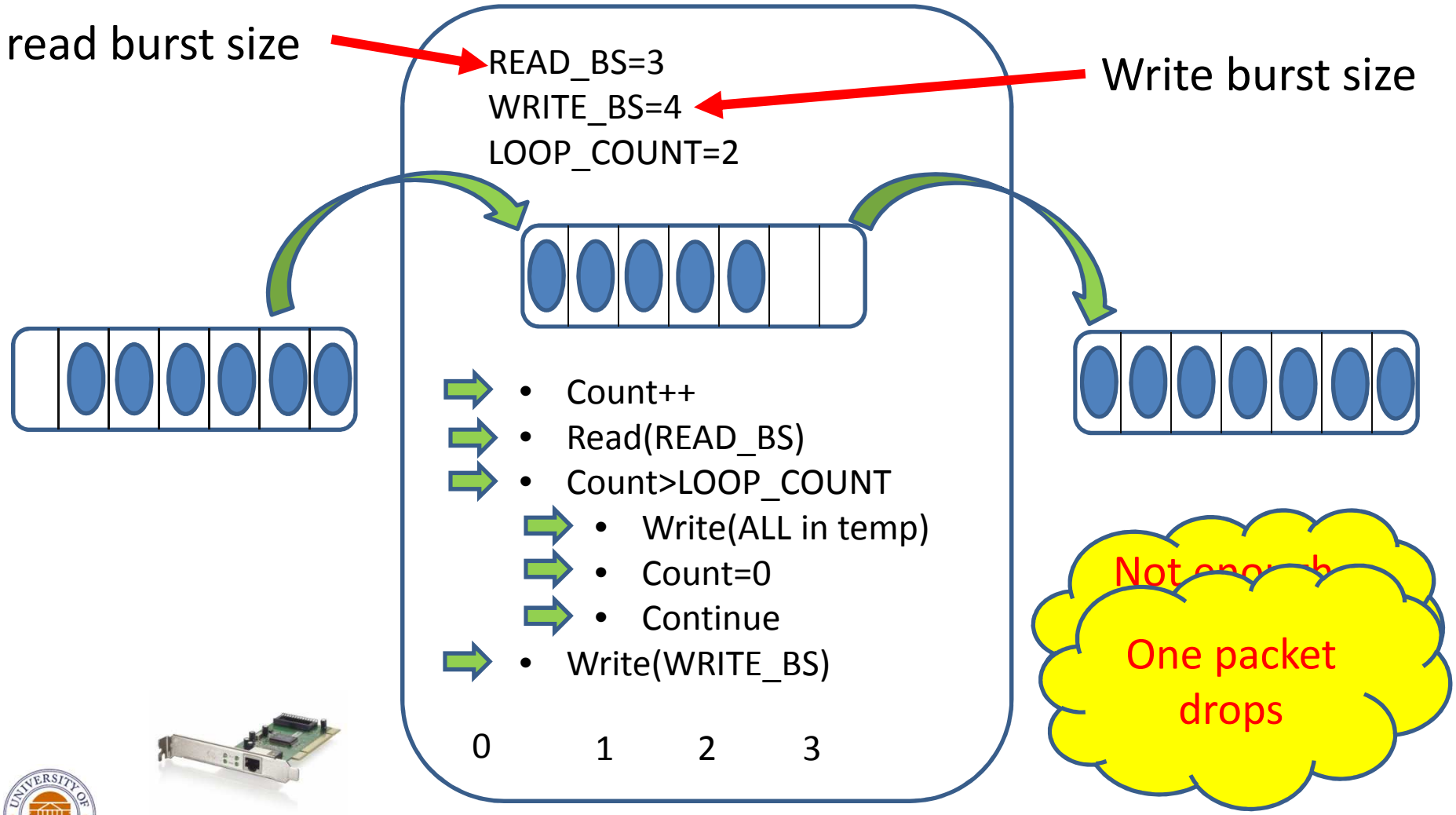
- High-speed software OpenFlow switch
  - Designed for multi-core processors
  - Leverages Intel's DPDK user-space drivers
- Key features
  - Parallelization rather than pipelining
  - Packet classification for load balancing
  - Packet coalescing
  - High-performance flow-table loops
    - Patricia trie and Hashing

Lagopus architecture (parallelization rather than pipelining)

Same code run by many worker threads



Packet coalescing



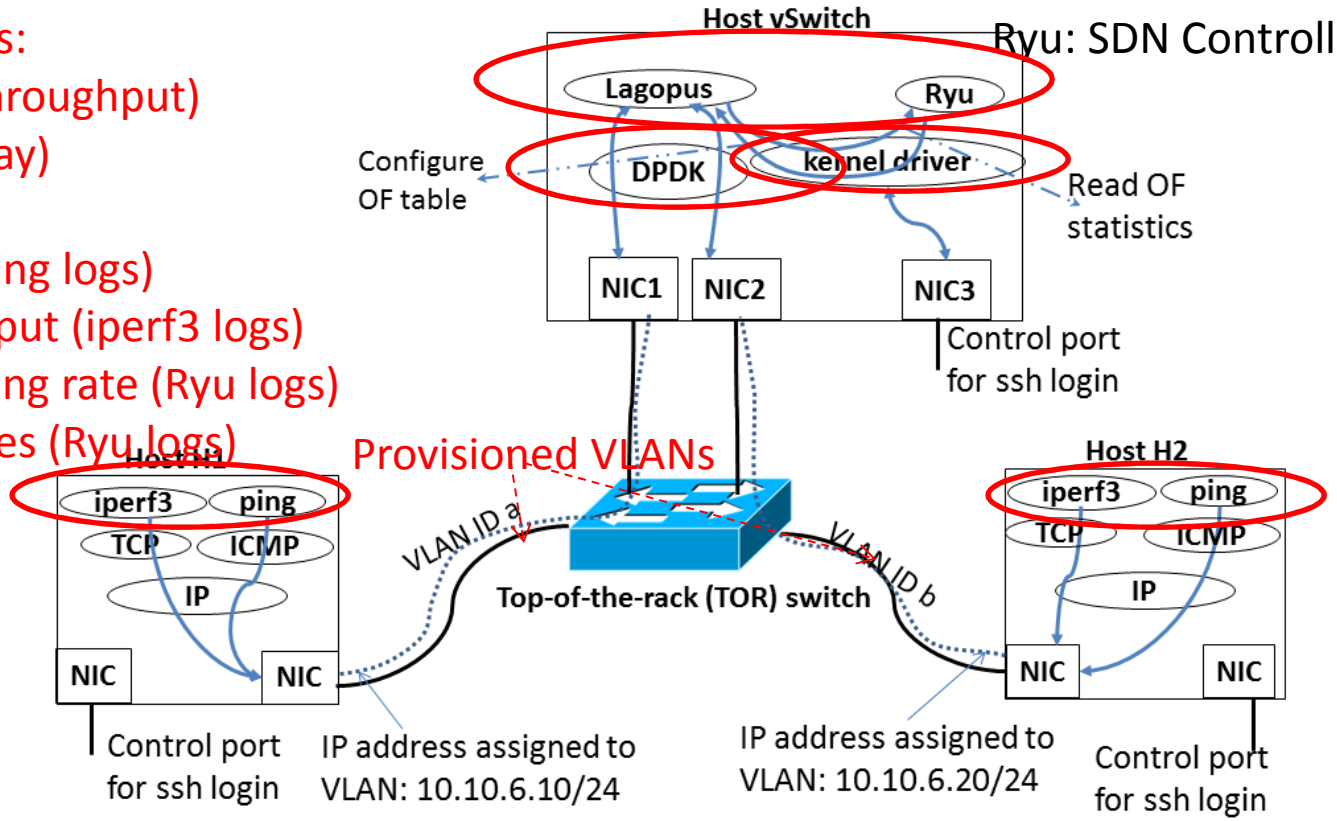
Experimental setups

- GENI setup
  - InstaGENI rack at U. Utah Downtown Data Center
  - Three bare-metal hosts (two 8-core CPUs)
  - 1 GE NICs compatible with Intel DPDK
- Keio U. setup
  - Two hosts (H1: 4-core; H2: two 10-core CPUs)
    - Loopback setup
  - 10 GE NICs (one compatible with Intel DPDK)
  - High-speed experiment

GENI experimental setup

Applications:
- iperf3 (throughput)
- ping (delay)

Metrics:
- delay (ping logs)
- throughput (iperf3 logs)
- forwarding rate (Ryu logs)
- drop rates (Ryu logs)



OF: OpenFlow

Lagopus/Ryu parameters

Parameter	Value
Number of receive threads	1
Number of transmit threads	1
Number of worker threads	$1, \cdots, 4$
I/O thread loop count	{100, 10000} packets
Worker thread loop count	1000 packets
Burst size (InstaGENI)	144 packets
Burst size (Keio)	32 packets
OpenFlow table size	$3, \cdots 1M$ entries
Statistics collection interval	10 seconds

- Varied three parameters

Experiments

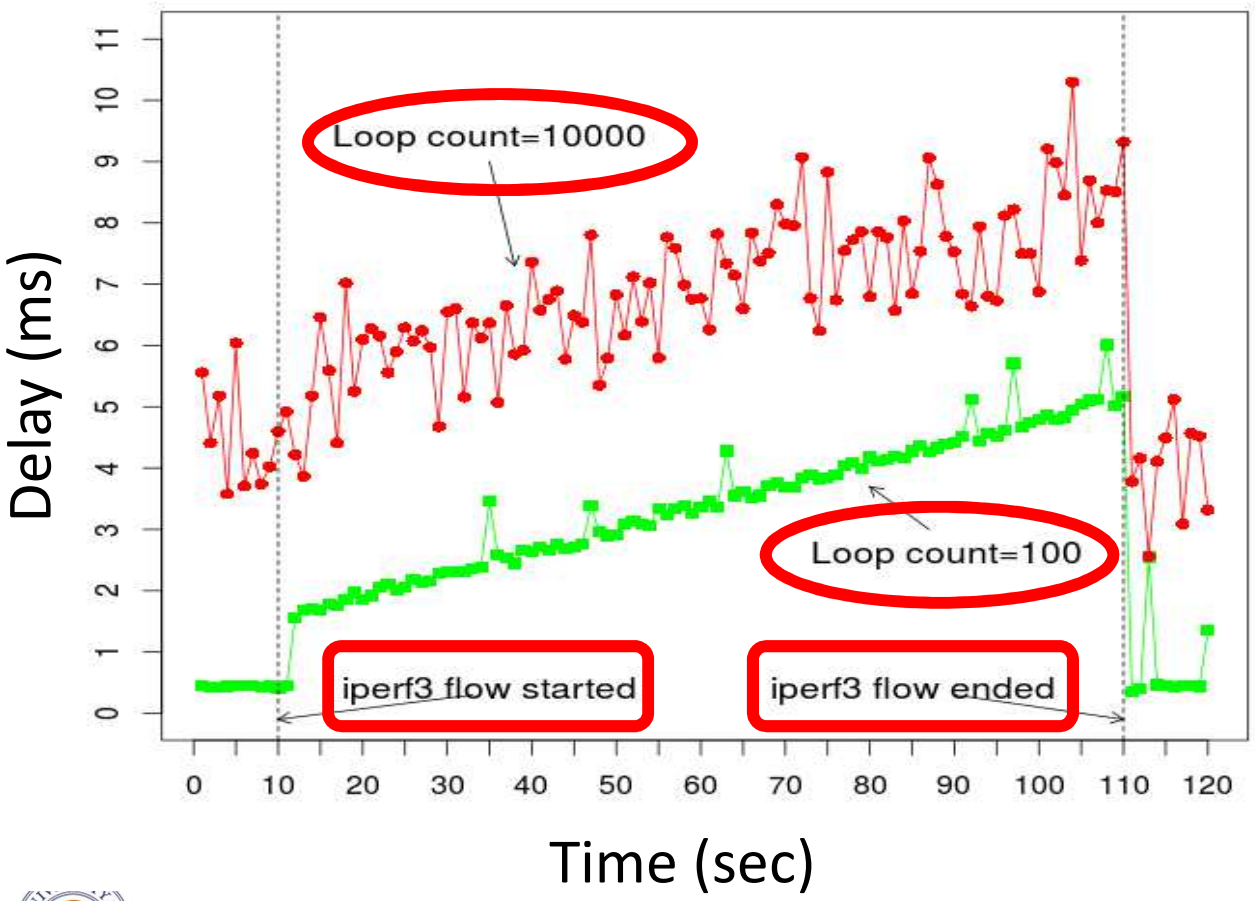
- InstaGENI testbed
  - Application performance
  - Impact of parallelization
- Keio testbed
  - High-throughput performance

ping-flow performance under low utilization

Loop count (pkts)	Application flows	Delay (ms)		Throughput (Mbps)	
		mean	SD	mean	SD
10000	`ping` only	4.468	1.159	NA	NA
100	`ping` only	0.507	0.447	NA	NA
10000	`iperf3` only	NA	NA	939.64	5.51
100	`iperf3` only	NA	NA	939.77	6.97
10000	both	See Fig. 3		939.51	5.70
100	both	See Fig. 3		939.09	5.57

- Time to fill 144-packet burst size > time to loop 10000 times
- With only ping flow, time to execute 10K loop was high, and hence ping delay was high $\Rightarrow$ low utilization: keep loop count small

Impact of iperf3 flow on ping delay



- ping and iperf3 flows sent to same worker thread; hence delay buildup
  - need receive-thread packet classification
- higher variance with 10K loop count: relative position of ping packet arrival within 10K loop

Impact of parallelization

No. of worker threads	No. of OF table entries	No. of flows	Flow (iperf3)	Throughput (Mbps) mean	SD
1	3	1	H1 → H2	939.5	6.9
1	3	2	H1 → H2	470	6.1
	3		H2 → H1	470	6.2
2	3	2	H1 → H2	935.58	5.74
	3		H2 → H1	935.54	5.4
1	1000	1	H1 → H2	427.5	12.8
1	1000	2	H1 → H2	184	36.1
	1000		H2 → H1	245.5	34.6
2	1000	2	H1 → H2	398.5	42.6
	1000		H2 → H1	390.4	40.13

- MAC address based OF table: Patricia trie not suitable
  - Even with just 1000 entries, load on worker threads was high
- Packet classification based on src-dst IP pairs
  - When sent in opposite direction, flows sent to different worker threads

High-performance tests (Keio U test bed; 10GE NICs)

Impact of the size of the OF table; pktgen used to create MPLS packets

- 1500B packets; Patricia trie works well

No. of OF table entries	No. of worker threads	Receive rate (Gbps)	Transmit (Forwarding) rate (Gbps)	Packet drop rate (%)
100K	1	9.810	9.808	0.002
100K	4	9.745	9.742	0.003
400K	1	9.813	9.809	0.03
400K	4	9.814	9.755	0.6
1M	1	9.823	9.797	0.27
1M	4	9.815	9.513	3.0

- Flow-based packet load balancing to worker threads was disabled
- Need to calibrate software switch for 0 packet drop rate
- Transmit thread finds more packets in ring buffer with 1 worker thread
- With 4 worker threads, fewer packets; so packet move cost not as amortized

Experimental contributions

- Most vSwitch performance evaluation centered on packet forwarding rates
- Our additional metrics:
  - Application performance: ping delays
  - Packet drop rates

Conclusions

- Need better worker thread load balancing with packet classification in receive threads
  - separate delay-sensitive packets from bulk-data flows
  - distribute flows based on more fields to different worker threads
- Packet coalescing loop count should be kept small when link utilization is low
- Calibrate switch for highest packet arrival rate at which drop rate is 0
- Transmit-side packet batching impacts choice of number of worker threads