ROYAL INSTITUTE OF TECHNOLOGY

OpenFlow with Intel 82599

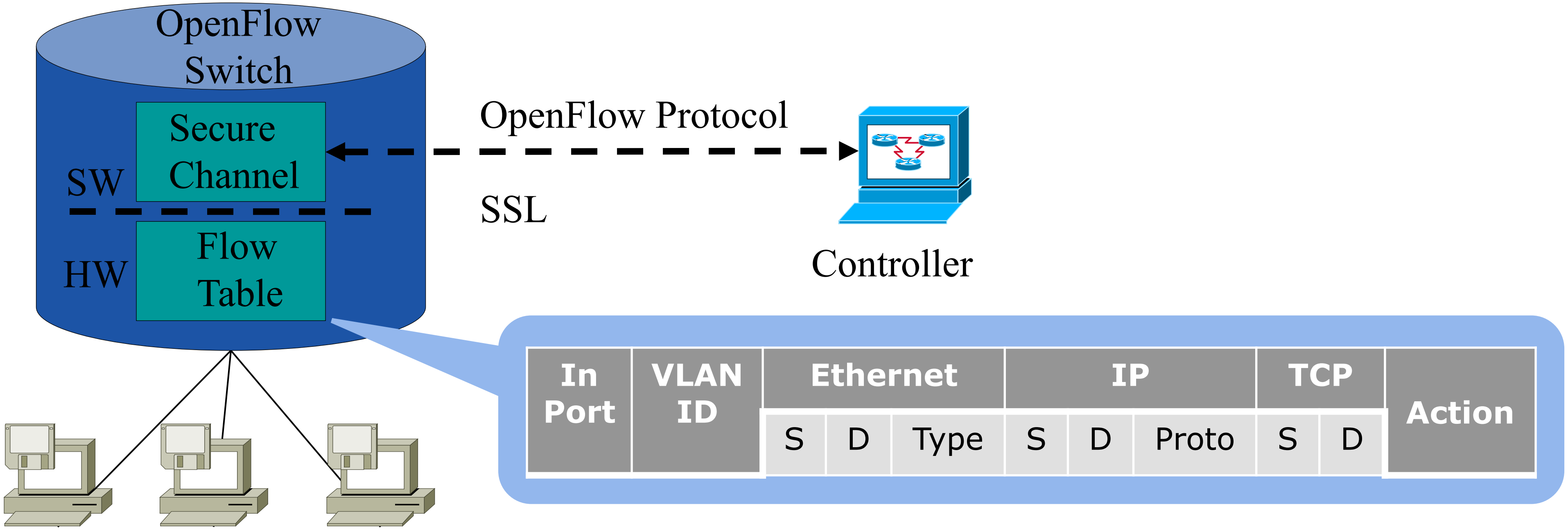
Voravit Tanyingyong, Markus Hidell, Peter Sjödin

Outline

- Background
- Goal
- Design
- Experiment and Evaluation
- Conclusion

OpenFlow

- Open up commercial network hardware for experiment
- Access to flow table without exposing internal workings
- Flow-based switching technology
  - Multiple fields in packet header
  - Wildcards on any fields

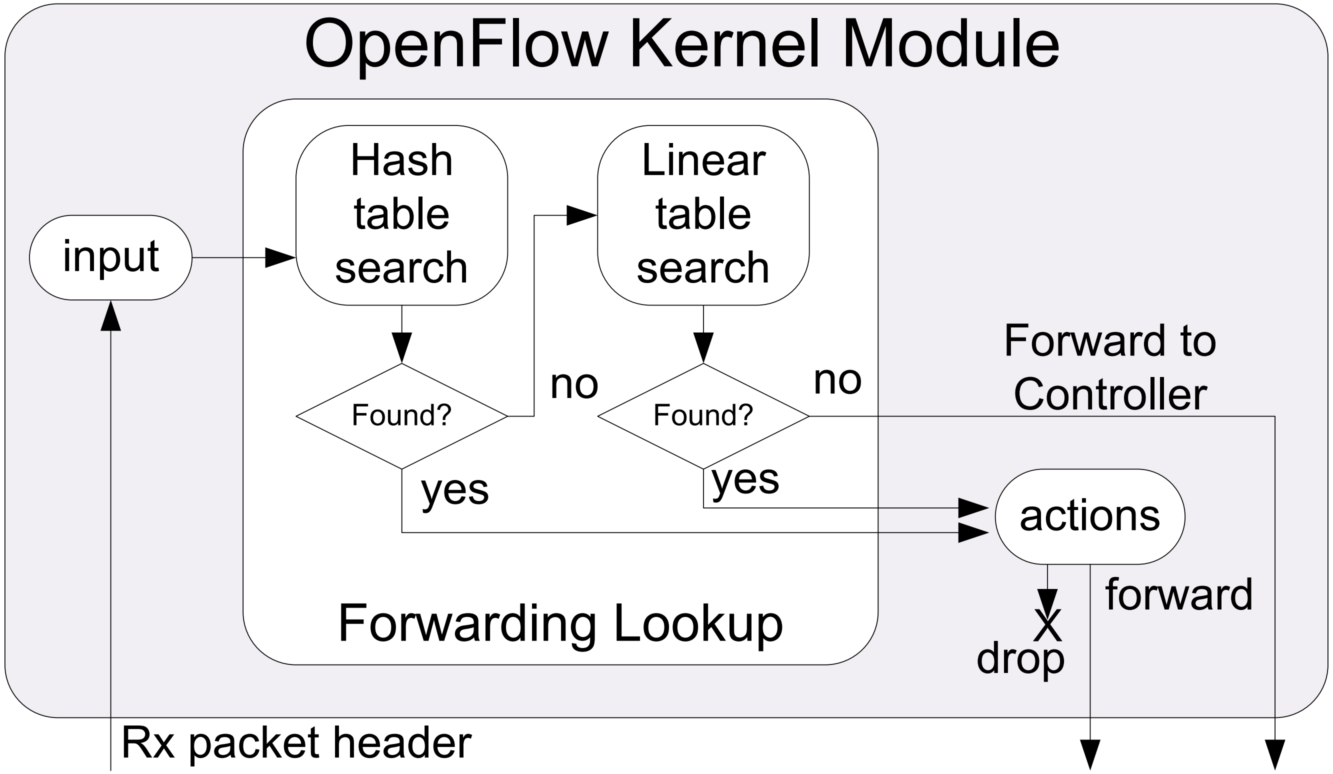


In Port	VLAN ID	Ethernet			IP			TCP		Action
		S	D	Type	S	D	Proto	S	D	

ROYAL INSTITUTE OF TECHNOLOGY

OpenFlow Software Lookup

- 2 lookup tables:
  - Hash: Exact match lookup
  - Linear: Wildcards lookup



Main goal

- Improve PC-based OpenFlow switching
  - OpenFlow lookup process
  - Open Source Software
  - Standard commodity hardware

- PC-based OpenFlow switch
  - OpenFlow kernel module
- Commodity NIC
  - Offload lookup from software

Offload Software Lookup to Hardware

- NIC with Intel 82599 10 GbE controller
  - Flow Director filters
  - Multiple receive queues
  - Receive-Side Scaling (RSS)

- Flow Director filters
  - Direct received packets to queues based on their flows
  - Filter covers:
    - VLAN header
    - IP: Source, Destination, IPv4/IPv6 protocol match
    - L4: Source, Destination, TCP/UDP/SCTP protocol match
    - Flexible 2-byte tuple anywhere in the first 64-byte of the packet header

Architectural Design

Port-mapping Table

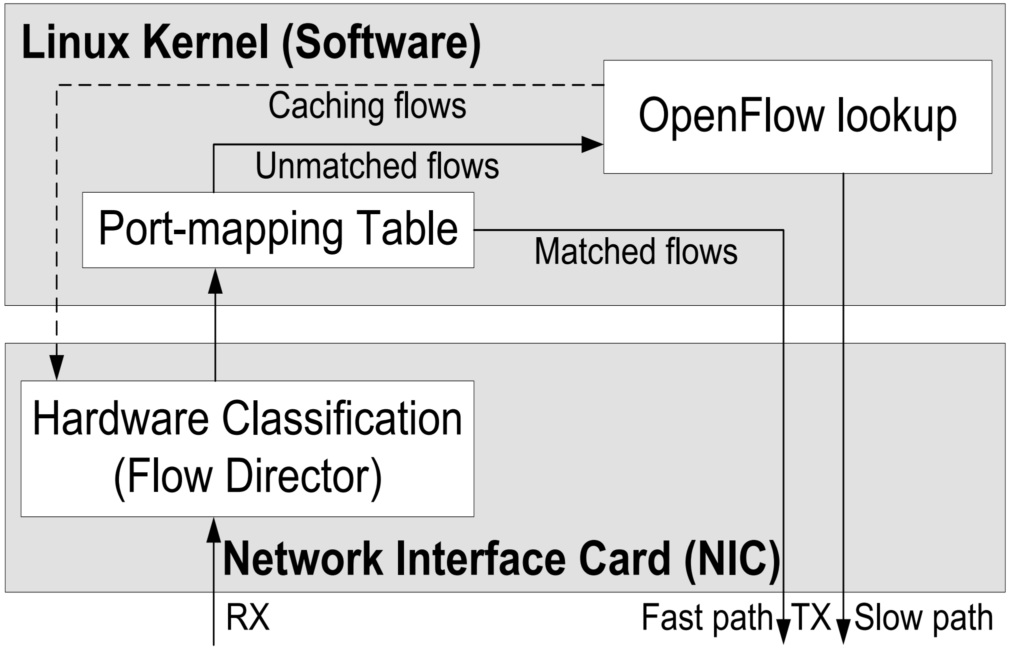
Input Port	Input Queue	Output Port	Output Queue
eth0	1	eth1	1
eth0	2	eth2	1

Map queue to output port

Offload lookup to NIC

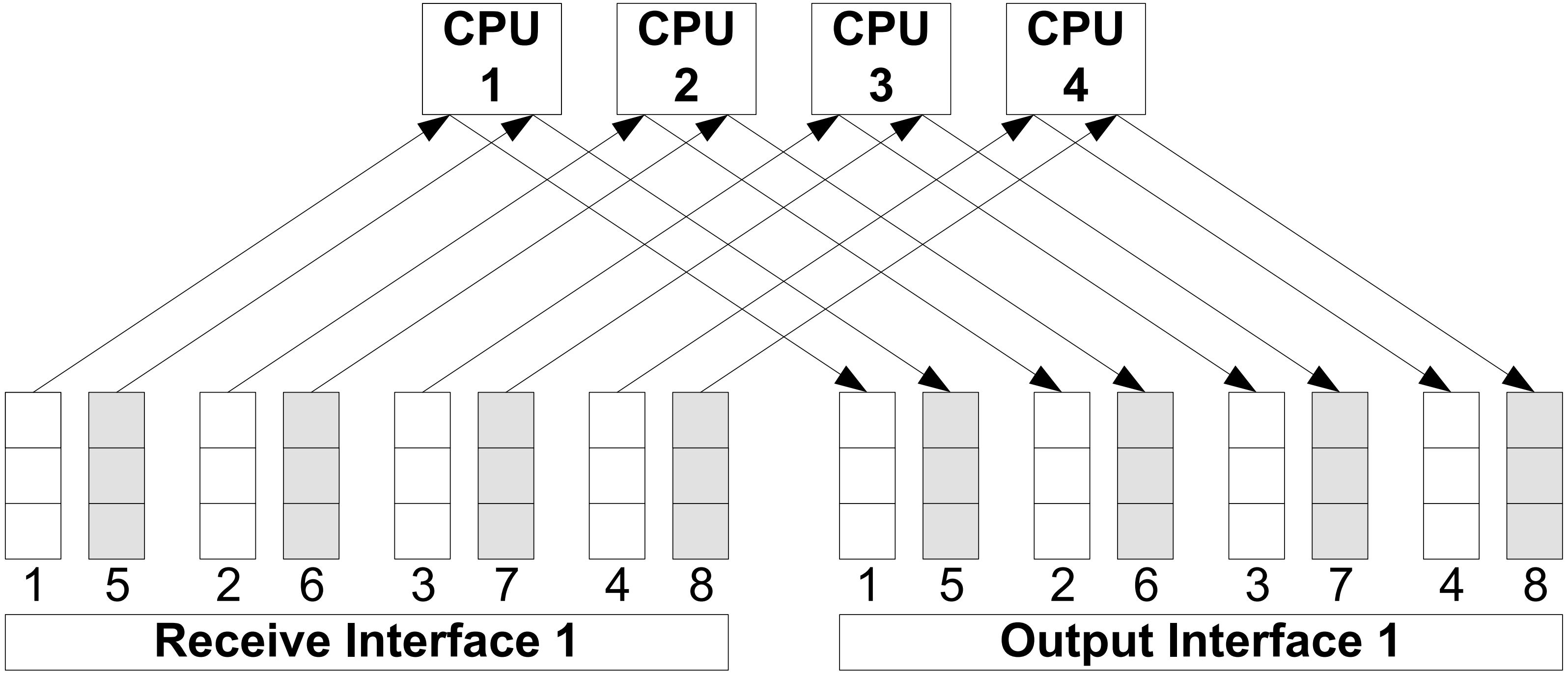
Packet header				Queue
VLAN	L2	IP	TCP	
Flow1 (src 1.1.1.1 dst 2.2.2.2)				1
Flow2 (src 2.2.2.1 dst 3.3.3.3)				2

Classifies packet to queue



Architectural Design (cont'd)

- Multi-queue and multi-core mappings



CPU 1

CPU 2

CPU 3

CPU 4

1 5 2 6 3 7 4 8

Receive Interface 1

1 5 2 6 3 7 4 8

Output Interface 1

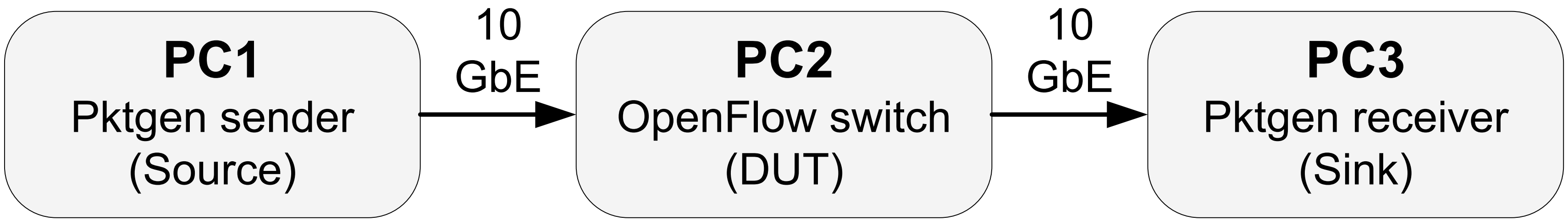
□ Queue for packets match a Flow Director filter

▨ Queue for unmatched packets (load-balance among CPU cores via RSS)

Experiment Setup

- Conformance with RFC 2544
- PC1: Traffic generator (Source)
  - Pktgen
- PC2: Device-Under-Test (DUT)
  - OpenFlow 0.8.9-rev4 kernel module
  - Intel NIC with Flow Director
- PC3: Traffic receiver (Sink)
  - Pktgen with receiver patch

- OS: Bifrost 6.1 with kernel net-next 2.6.34-rc2
- Hardware: standard components
  - Motherboard: TYAN S7002
  - CPU: Intel Xeon Quad Core 5520, 2.26 GHz
  - RAM: 3x1GB REG ECC
  - PSU: 460 watt power
  - NIC: Intel Ethernet Server Adapter X520-SR2



Experiment Scenarios

- Baseline Performance Test
- Modification Overhead
- Flow Caching
- Multi-core Scaling

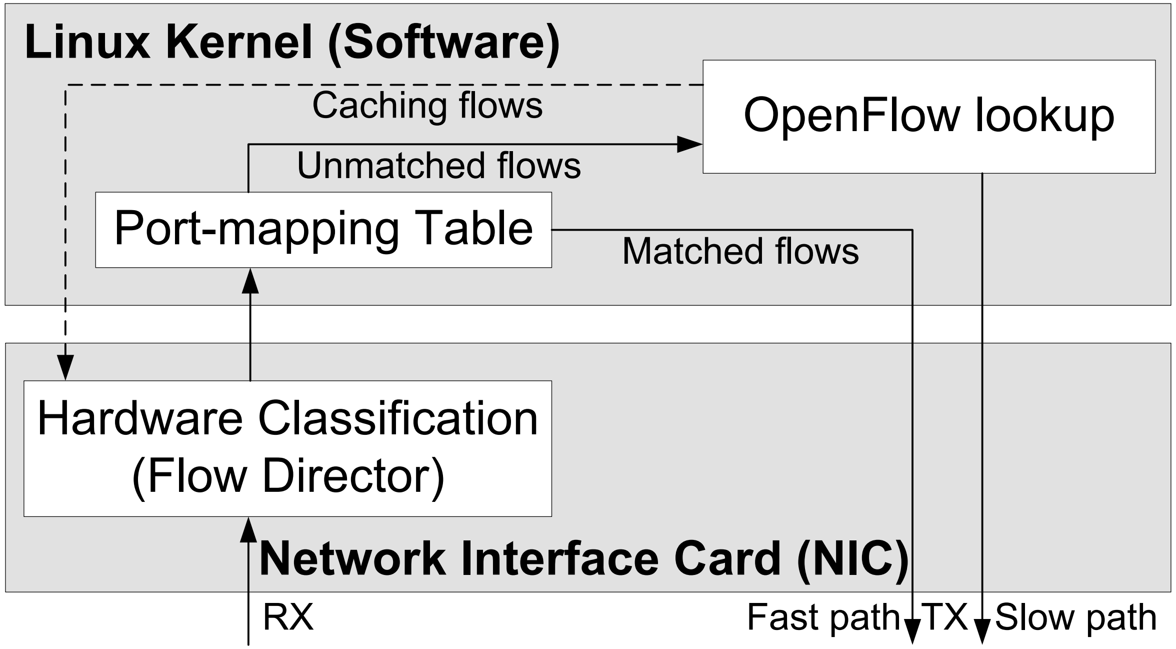
Baseline Performance Test

- Performance increases 41.50% on average
- Independent of table size

# Entries	Linear (in kpps)	Hash (in kpps)	Port-mapping (in kpps)
1	523	-	814
100	360	581	816
1100	82	578	817
2100	45	577	816
3100	31	577	816
4100	24	576	816
5100	20	576	817
6100	16	576	818
7100	14	576	816
8100	13	575	815

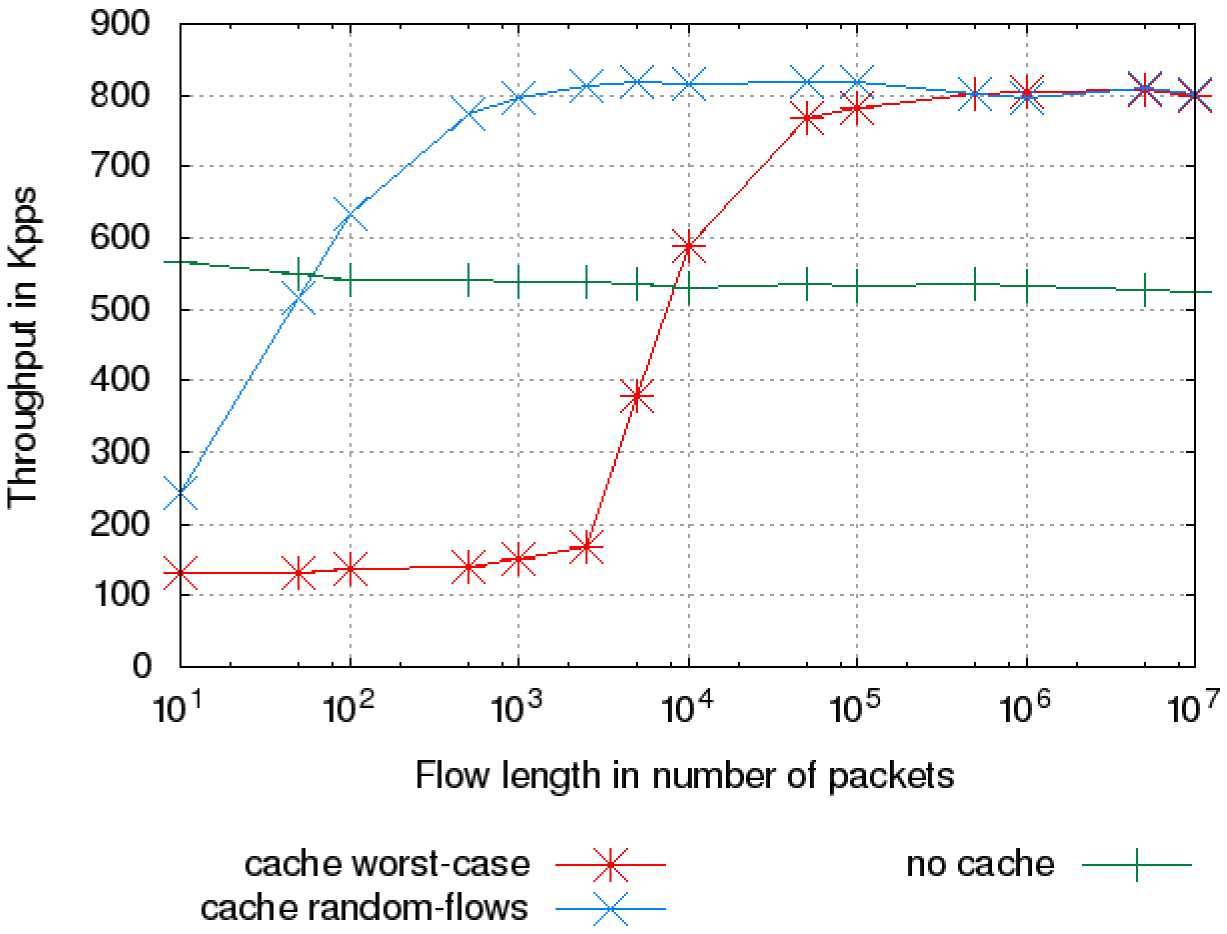
Modification Overhead

- Overhead: no match found in Port-mapping table
- Negilible overhead (on average 0.40%)



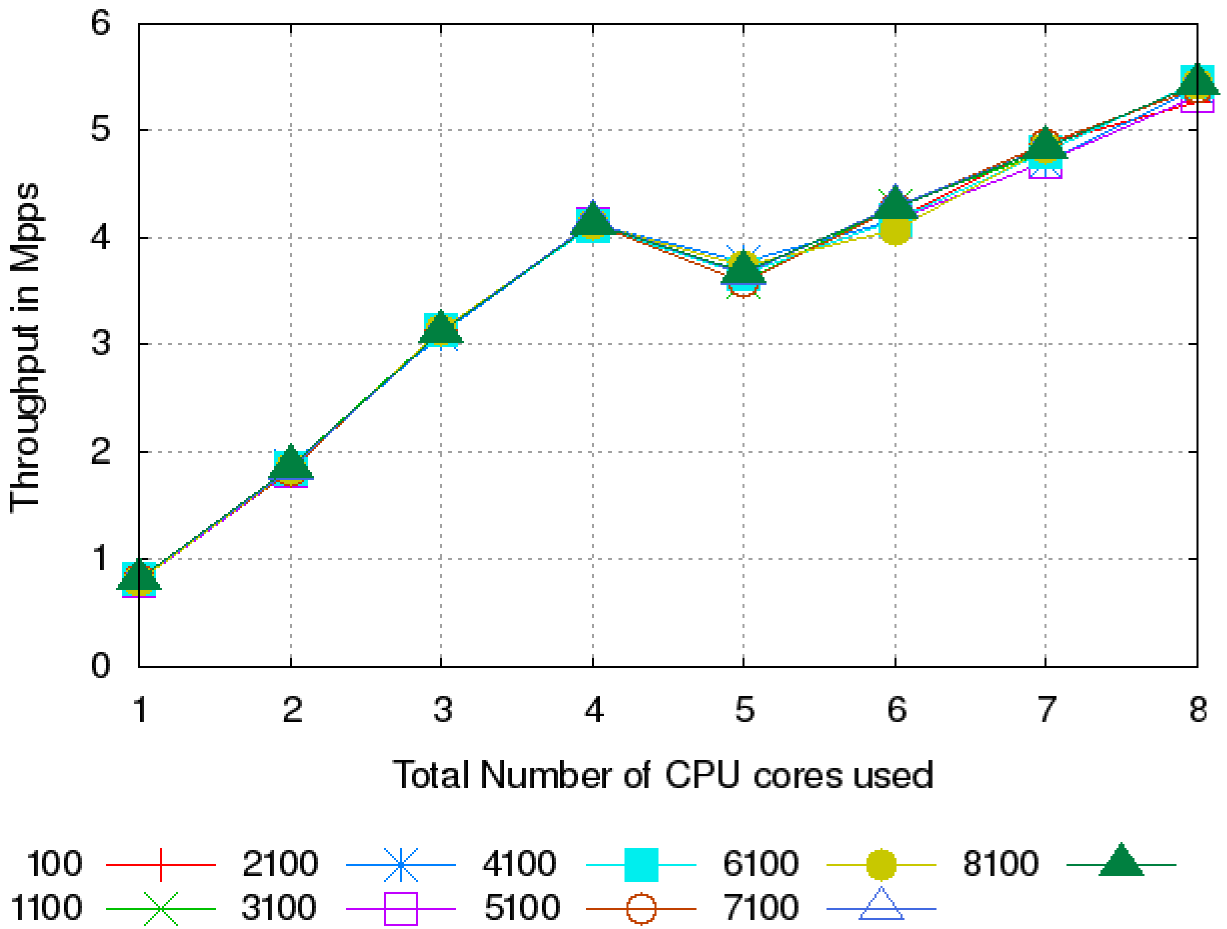
Flow Caching

- How flow length affects the performance



Multi-core Scaling

- Hyper-Threading has adverse effects on throughput



Tau Leng et al. - An empirical study of hyper-threading in high performance computing clusters

Conclusions and Future Work

- Our architecture significantly improves lookup performance of PC-based OpenFlow switch
- Negligible overhead
- The cost to cache a flow need to be amortized over
  - 100 packets in randomized flows case
  - 10000 packets in the worst case.
- Scale with multicore system
- Adverse effects from Hyper-Threading

Conclusions and Future Work (cont'd)

- Flow Director has promising potential for flow caching
- Tweaking different parameters on the NIC such as traffic classes, packet buffer size can enhance our architecture
- Adding physical CPU, memory together with NUMA-friendly configuration can enhance our architecture
- Further study on more advanced caching scheme suitable for actual network characteristic

KTH VETENSKAP OCH KONST

ROYAL INSTITUTE OF TECHNOLOGY

Thank You for Listening!