TECHNION
Israel Institute
of Technology

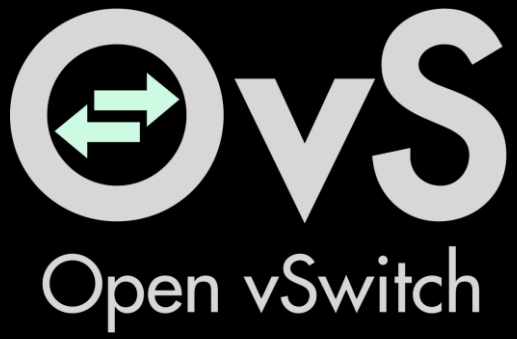
ACSL
Accelerated Computing
Systems Lab

Scaling Open vSwitch with a Computational Cache

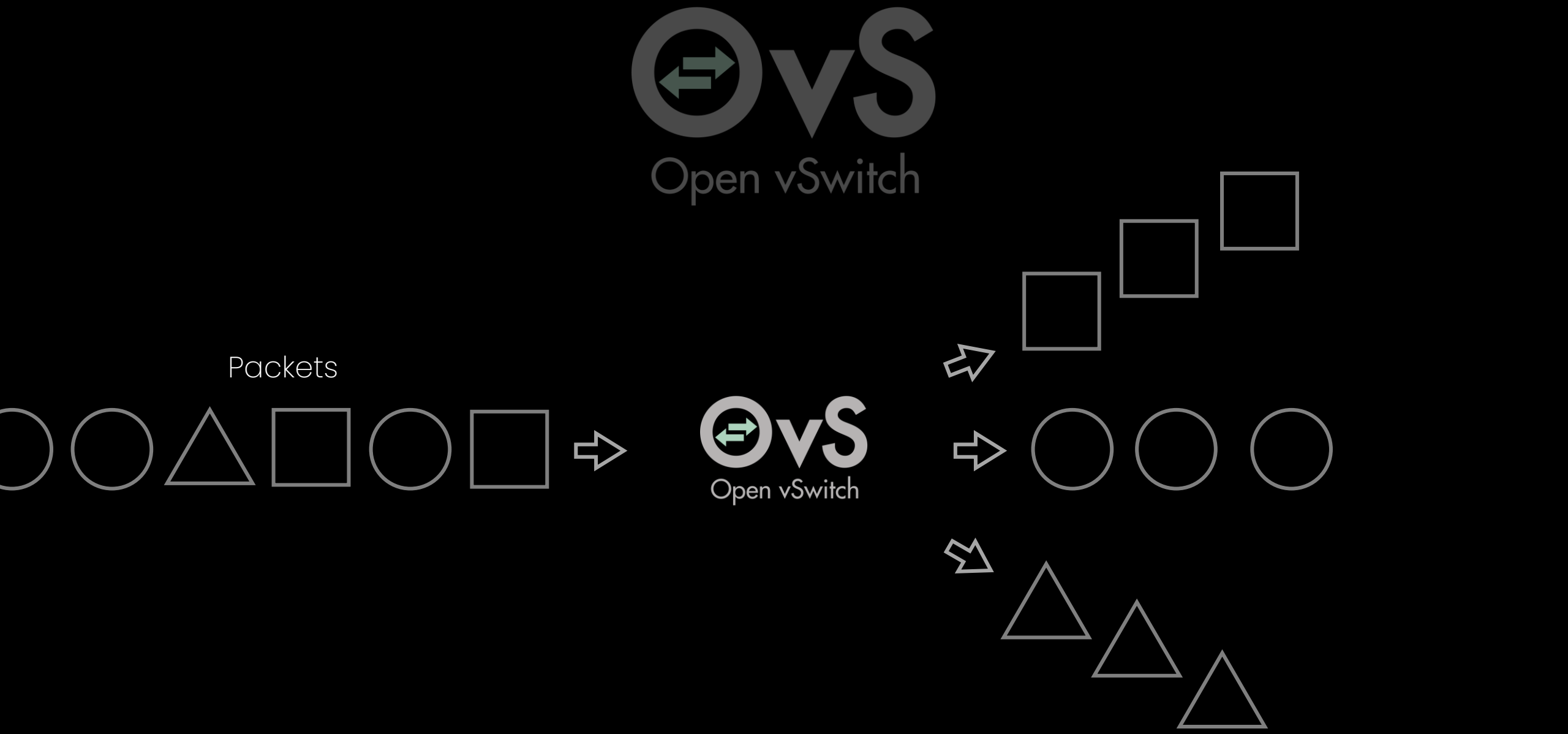
Alon Rashelbach Ori Rottenstreich Mark Silberstein

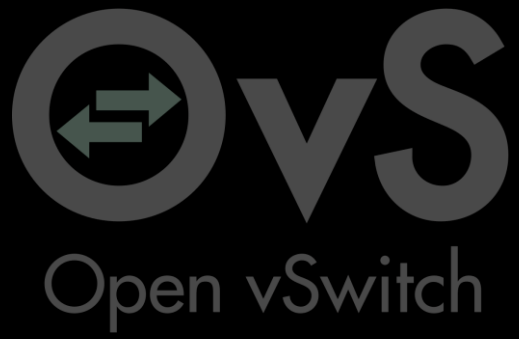
Technion, Israel

USENIX NSDI 2022

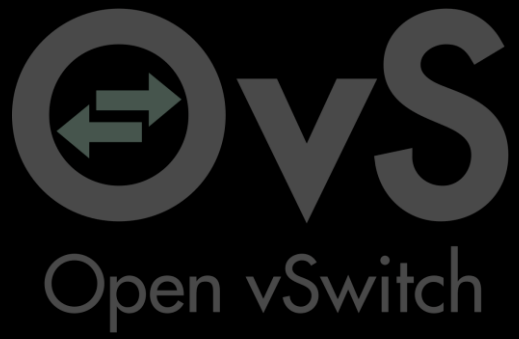


- Virtual Switch
- Open-source
- Data-centers


OvS
Open vSwitch
Packets
OvS
Open vSwitch



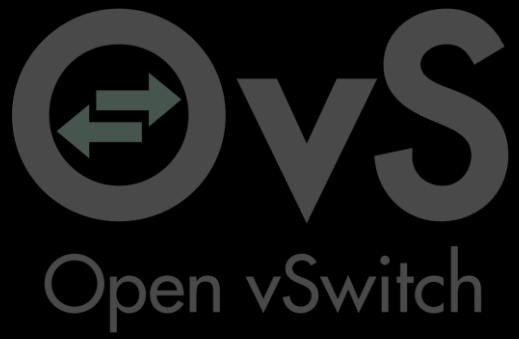
1K OpenFlow Rules ⇨ 500K OpenFlow Rules



1K OpenFlow Rules ⇨ 500K OpenFlow Rules

14x geomean throughput degradation

In some cases: throughput drops from a few Mpps to a few Kpps

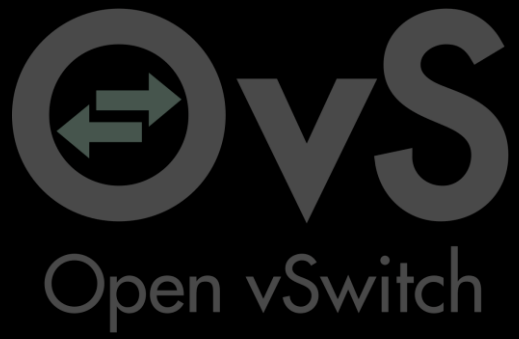


1K OpenFlow Rules ⇨ 500K OpenFlow Rules

14x geomean throughput degradation

In some cases: throughput drops from a few Mpps to a few Kpps

In this paper: a geomean speedup of 12x



1K OpenFlow Rules ⇨ 500K OpenFlow Rules

14x geomean throughput degradation

In some cases: throughput drops from a few Mpps to a few Kpps

In this paper: a geomean speedup of 12x

Neural-net inference per packet in OVS data-path

Background

Challenge

This paper

Integration with OVS

Evaluation

NuevoMatch algorithm for packet classification

(*) Rashelbach et al., "A Computational Approach to Packet Classification". *SIGCOMM 2020*.

NuevoMatch algorithm for packet classification

OpenFlow rules

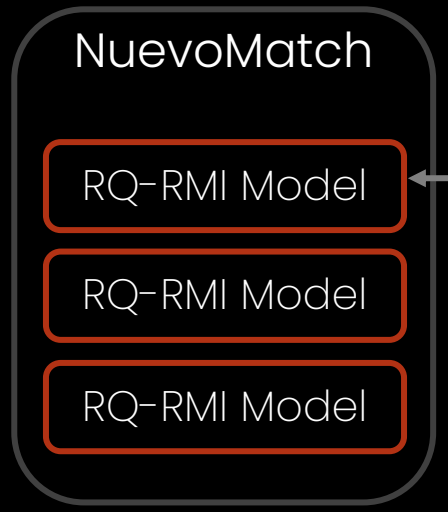
Src IP	Dst IP	Action
`10.0.10.*`	`56.0.*.*`	`Port 1`
`10.0.*.*`	`56.22.7.1`	`Drop`

(*) Rashelbach et al., "A Computational Approach to Packet Classification". *SIGCOMM 2020*.

NuevoMatch algorithm for packet classification

OpenFlow rules

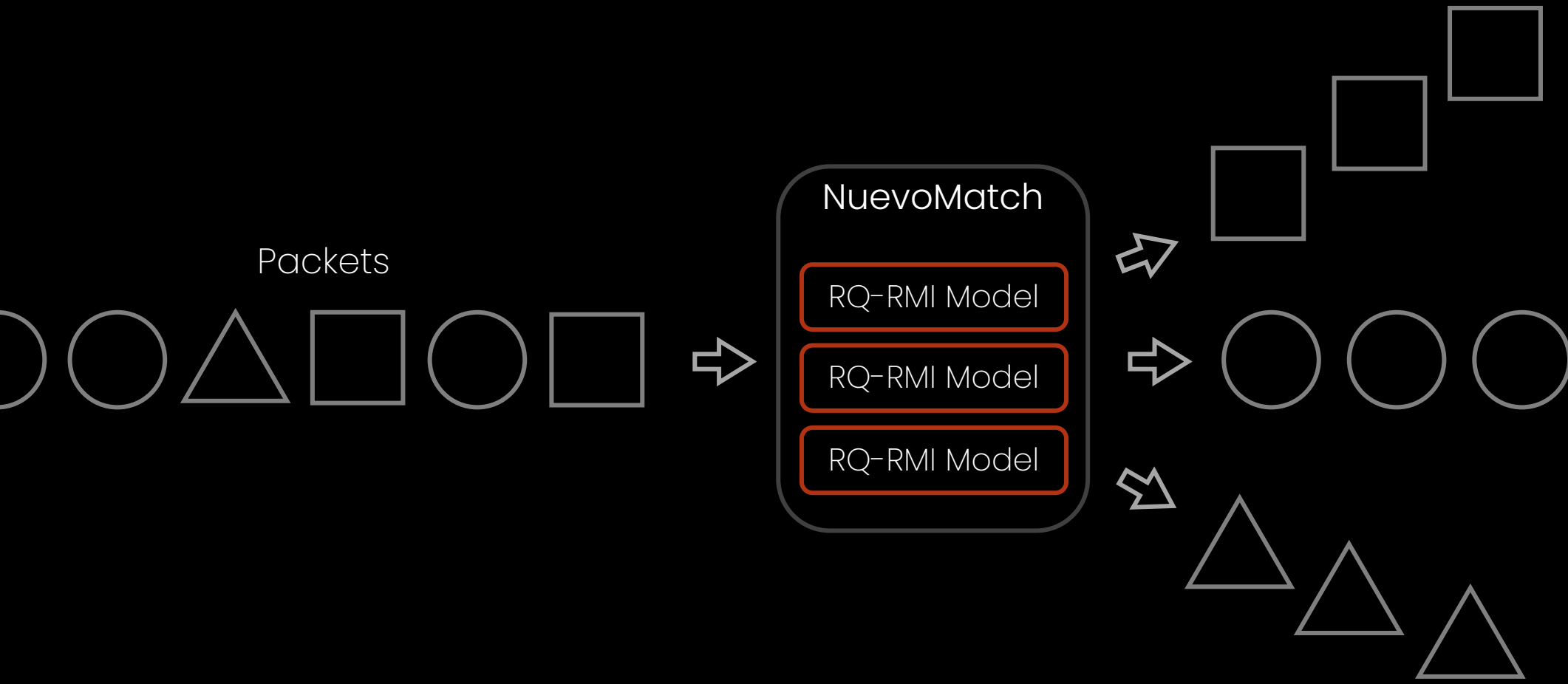
Src IP	Dst IP	Action
10.0.10.*	56.0.*.*	Port 1
10.0.*.*	56.22.7.1	Drop



Unique model architecture
40ns NN inference on CPU

(*) Rashelbach et al., "A Computational Approach to Packet Classification". *SIGCOMM 2020*.

NuevoMatch algorithm for packet classification

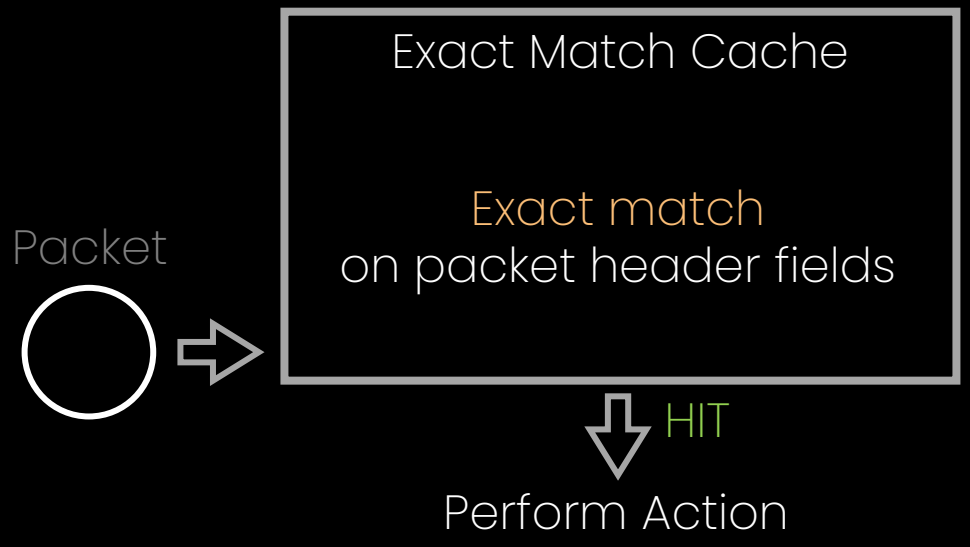


Classifies packets on the CPU using Neural Network Inference

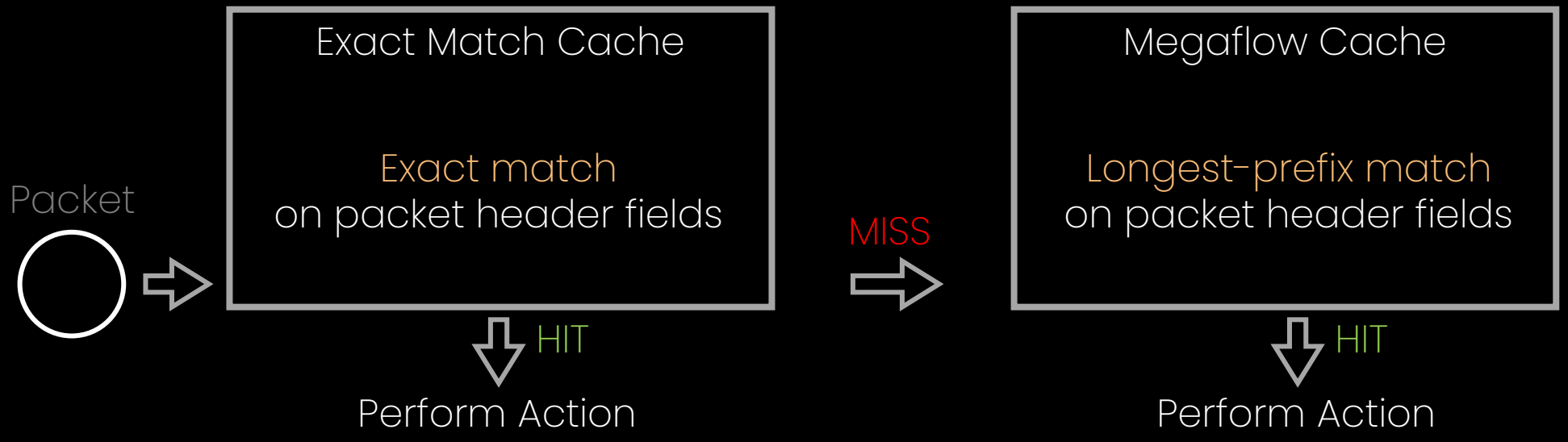
(*) Rashelbach et al., "A Computational Approach to Packet Classification". *SIGCOMM 2020*.

OVS Data-path

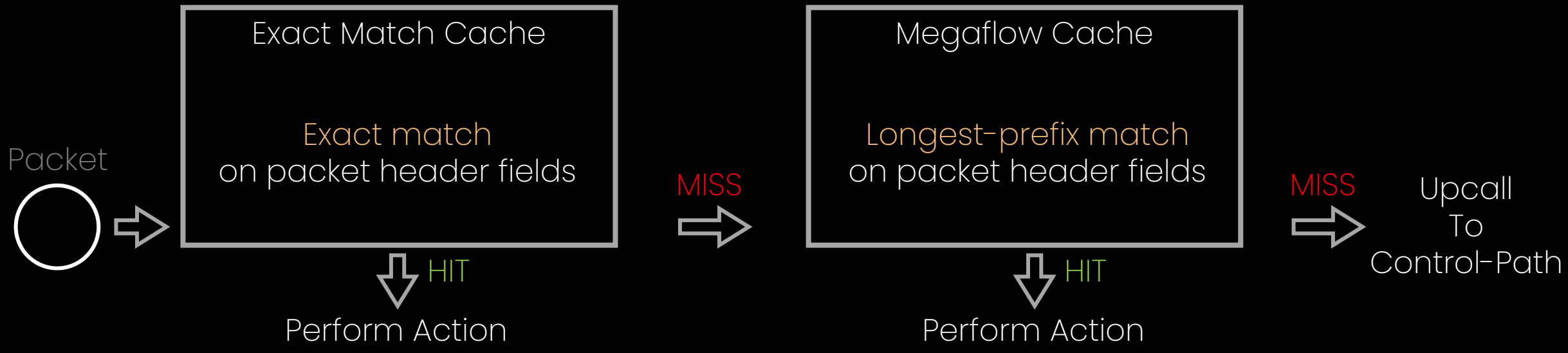
OVS Data-path



OVS Data-path



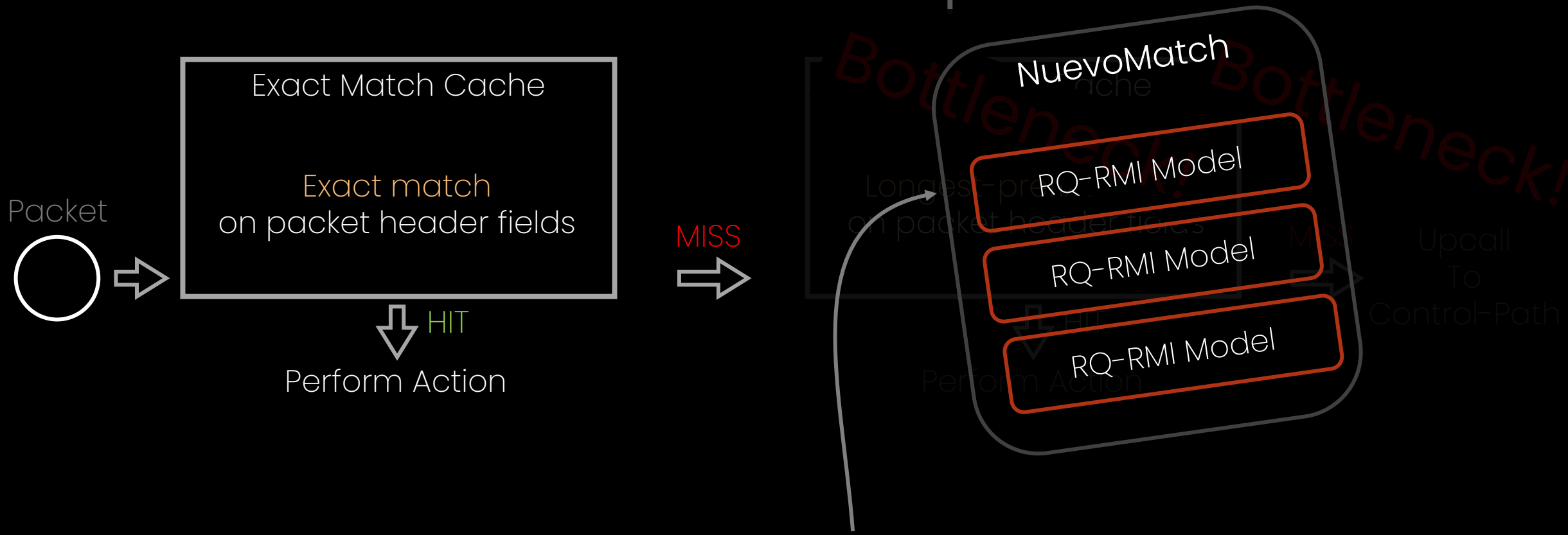
OVS Data-path



OVS Data-path

See paper for details

OVS Data-path



Can we use the **NuevoMatch** algorithm within the **OVS** data-path?

NO!

Can we use the **NuevoMatch** algorithm
within the **OVS** data-path?

Challenge: Slow Updates

Challenge: Slow Updates

NuevoMatch uses neural-nets

to learn rules

Neural-nets must be trained

when rules change

The training time depends

on the total number of rules

Challenge: Slow Updates

NuevoMatch uses neural-nets

to learn rules

Neural-nets must be trained

when rules change

The training time depends

on the total number of rules

Train once

Challenge: Slow Updates

NuevoMatch uses neural-nets

to learn rules

Neural-nets must be trained

when rules change

The training time depends

on the total number of rules

Train fast

Must Support Frequent Updates

100K updates per second (on average)

(*) Katsikas et al. What you need to know about (smart) network interface cards. PAM, 2021.

Must Support Frequent Updates

100K updates per second (on average)

Training an RQ-RMI model with 100K+ rules
takes 5-20 minutes!

(*) Katsikas et al. What you need to know about (smart) network interface cards. PAM, 2021.

Must Support Frequent Updates

100K updates per **second** (on average)

Training an RQ-RMI model with 100K+ rules takes **5-20 minutes!**

NuevoMatch update rate is 1000x slower!

(*) Katsikas et al. What you need to know about (smart) network interface cards. PAM, 2021.

Background: RQ-RMI models in NuevoMatch

Range: a single rule field

Row num	Range
1	0-100
2	130-700
3	800-803
4	807-890
5	970-990
6	991-991

Background: RQ-RMI models in NuevoMatch

Range: a single rule field

Row num	Range
1	0-100
2	130-700
3	800-803
4	807-890
5	970-990
6	991-991

Sample the table for generating [range, #row] *tuples on the fly*

Background: RQ-RMI models in NuevoMatch

Range: a single rule field

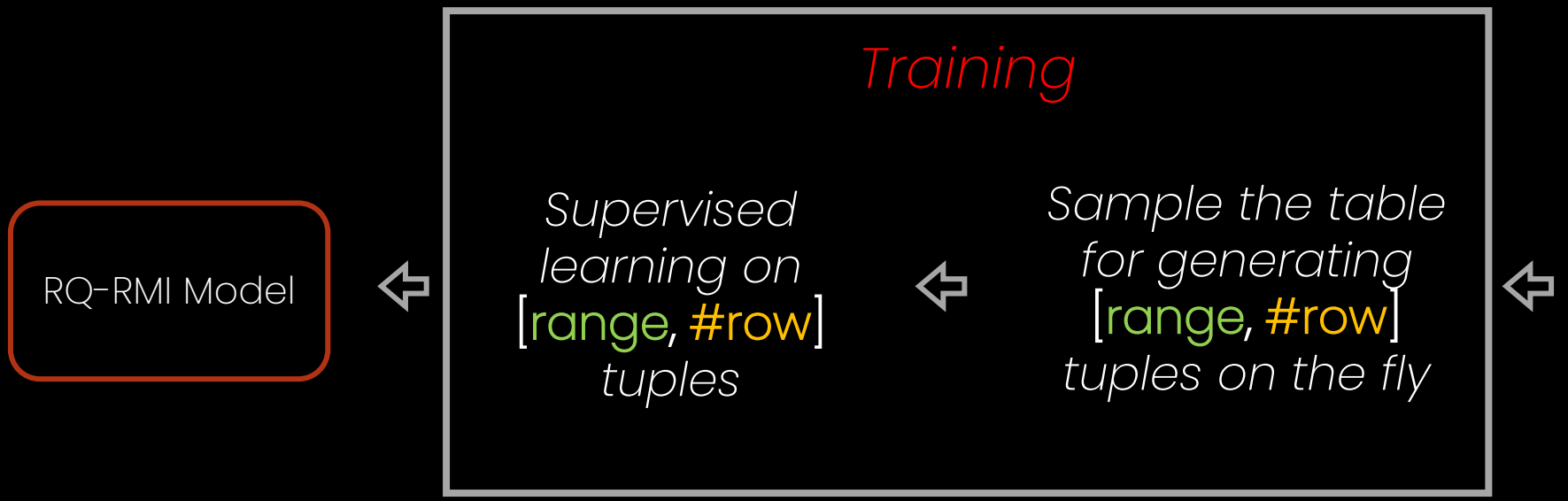
Row num	Range
1	0-100
2	130-700
3	800-803
4	807-890
5	970-990
6	991-991

Sample the table for generating [range, #row] *tuples on the fly*

Supervised learning on [range, #row] *tuples*

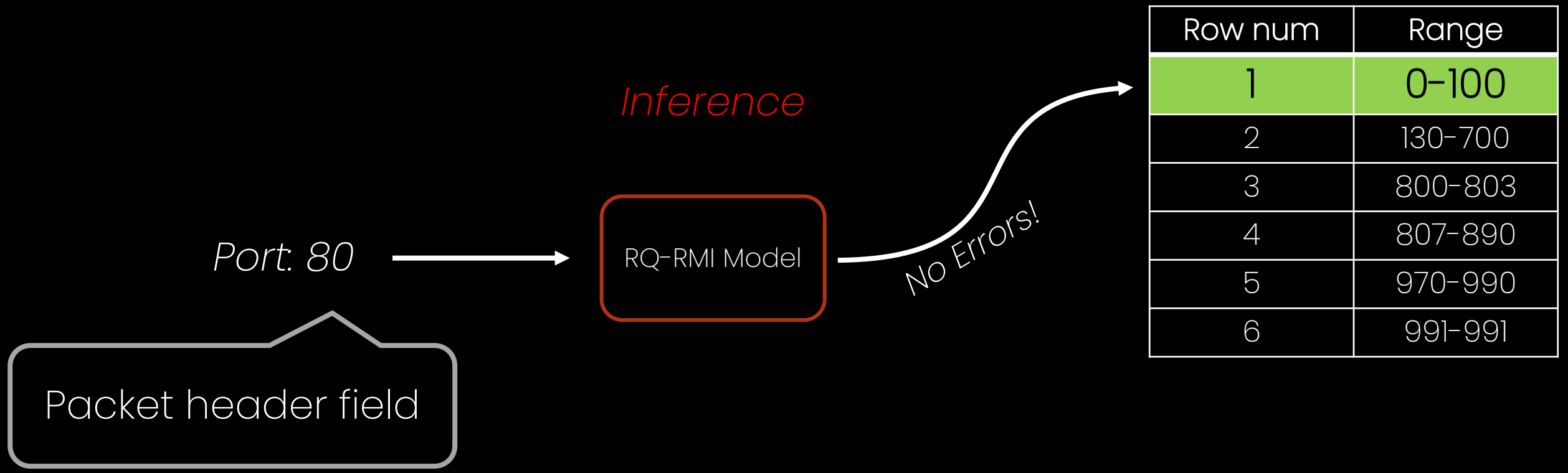
RQ-RMI Model

Background: RQ-RMI models in NuevoMatch



Row num	Range
1	0-100
2	130-700
3	800-803
4	807-890
5	970-990
6	991-991

Background: RQ-RMI models in NuevoMatch



Row num	Range
1	0–100
2	130–700
3	800–803
4	807–890
5	970–990
6	991–991

Background

Challenge

This paper

Integration with OVS

Evaluation

introducing *NuevoMatch**UP***

speeding-up NuevoMatch updates

introducing *NuevoMatch**UP***

speeding-up NuevoMatch updates

Three orthogonal improvements:

Bucketizing Ranges

Efficient SIMD implementation

Optimized sampling (see paper)

introducing *NuevoMatch**UP***

speeding-up NuevoMatch updates

Bucketizing Ranges

Row num	Range
1	0–10
2	20–30
3	40–50
4	60–70
5	80–90
6	100–110

introducing *NuevoMatch**UP***

speeding-up NuevoMatch updates

Bucketizing Ranges

Row num	Bucket	Ranges
1	0–30	0–10 20–30
2	40–70	40–50 60–70
3	80–110	80–90 100–110

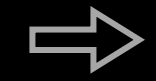
Row num	Range
1	0–10
2	20–30
3	40–50
4	60–70
5	80–90
6	100–110

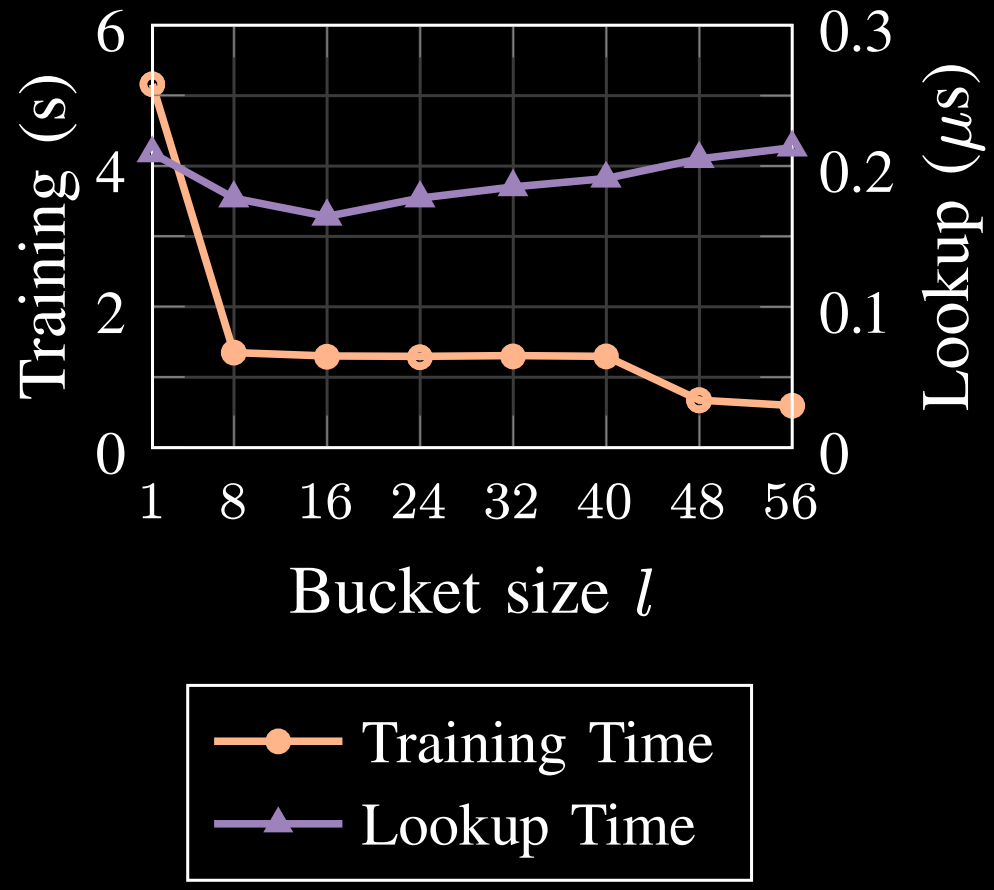
introducing *NuevoMatch**UP***

speeding-up NuevoMatch updates

Bucketizing Ranges

Row num	Bucket	Ranges
1	0-30	0-10 20-30
2	40-70	40-50 60-70
3	80-110	80-90 100-110



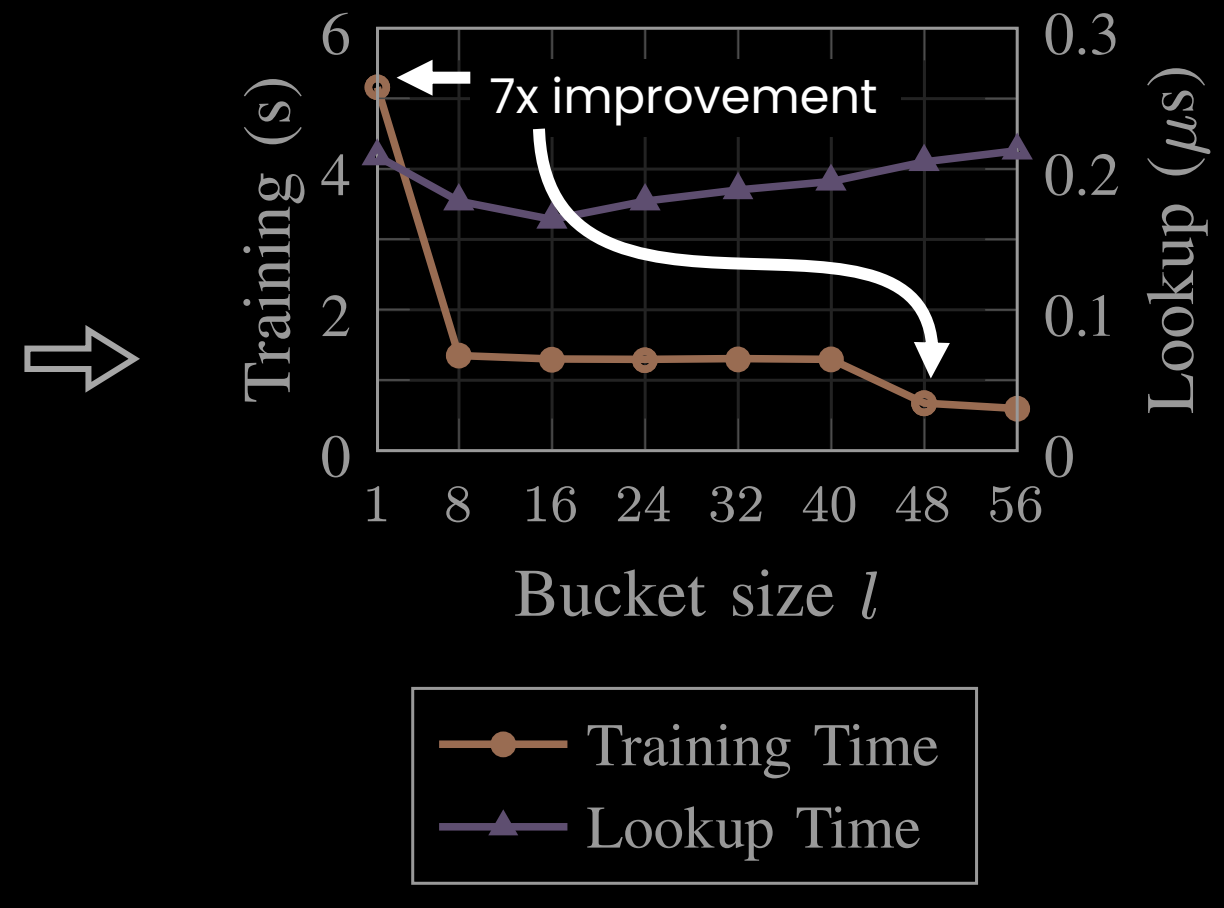


introducing *NuevoMatch**UP***

speeding-up NuevoMatch updates

Bucketizing Ranges

Row num	Bucket	Ranges
1	0-30	0-10 20-30
2	40-70	40-50 60-70
3	80-110	80-90 100-110

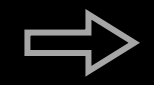


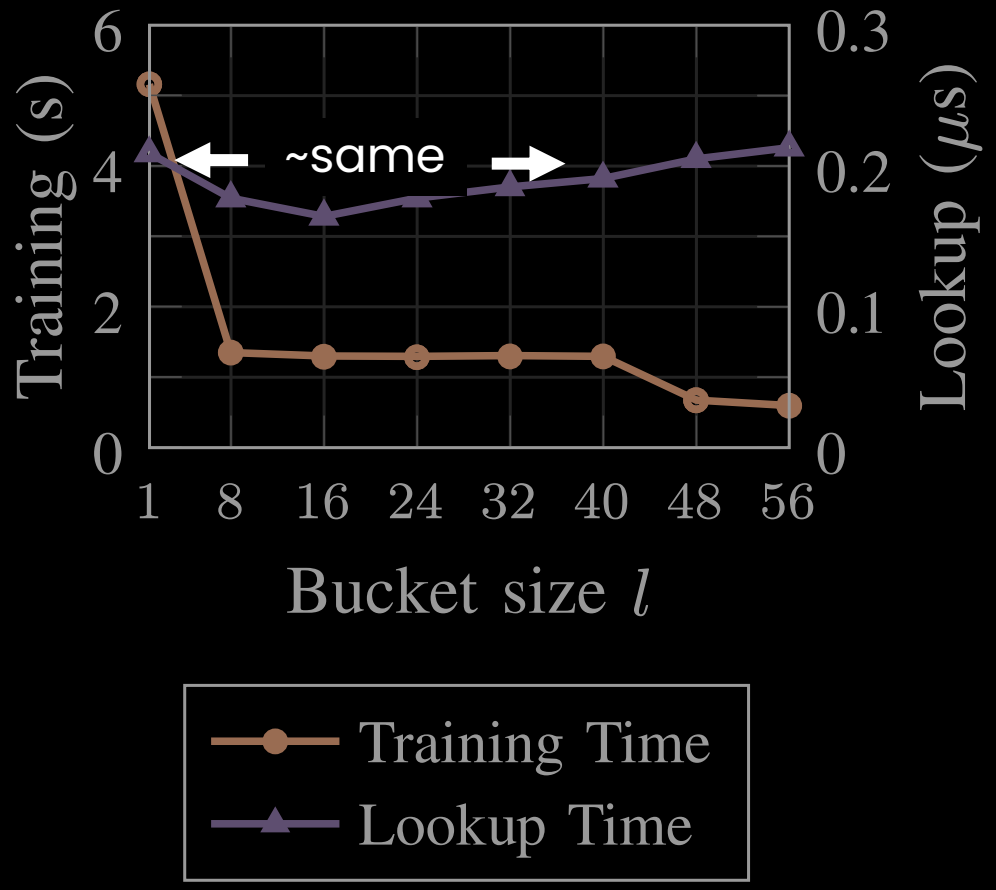
introducing *NuevoMatch**UP***

speeding-up NuevoMatch updates

Bucketizing Ranges

Row num	Bucket	Ranges
1	0-30	0-10 20-30
2	40-70	40-50 60-70
3	80-110	80-90 100-110





introducing *NuevoMatch**UP***

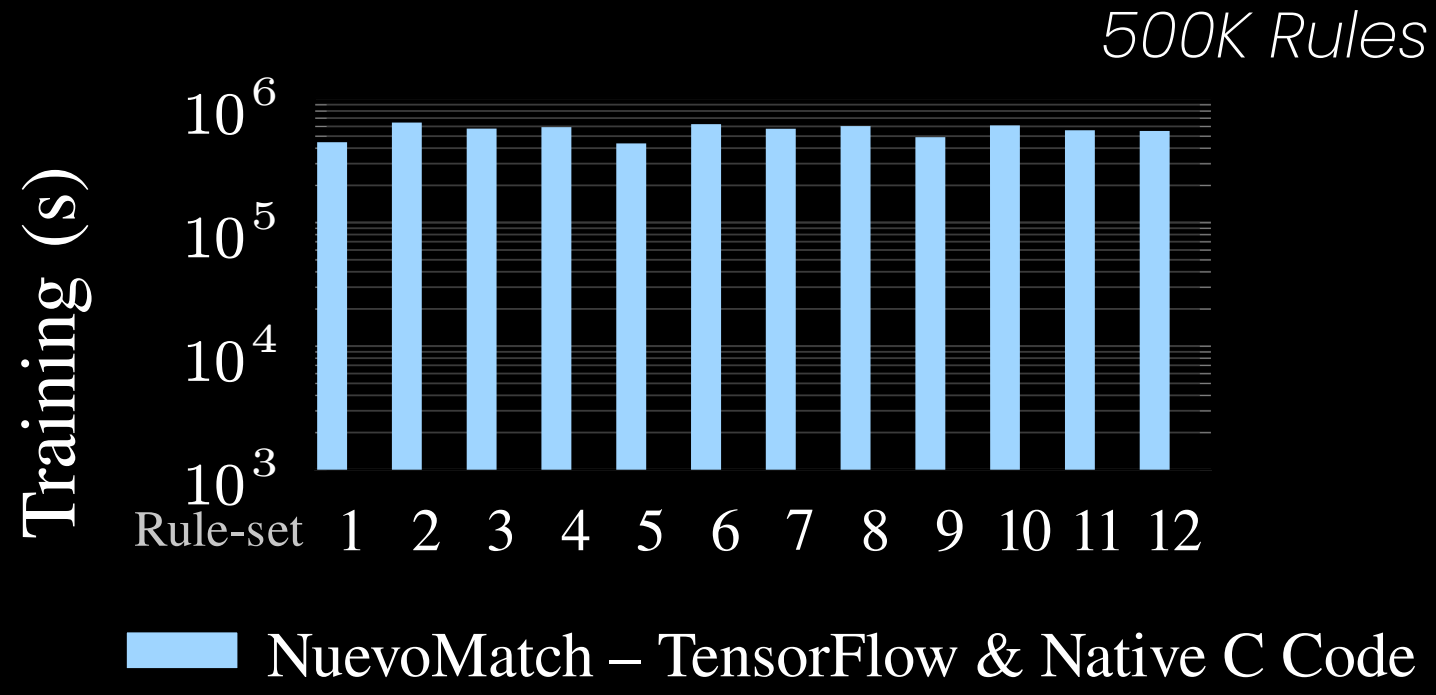
speeding-up NuevoMatch updates

Efficient SIMD Implementation

introducing *NuevoMatch**UP***

speeding-up NuevoMatch updates

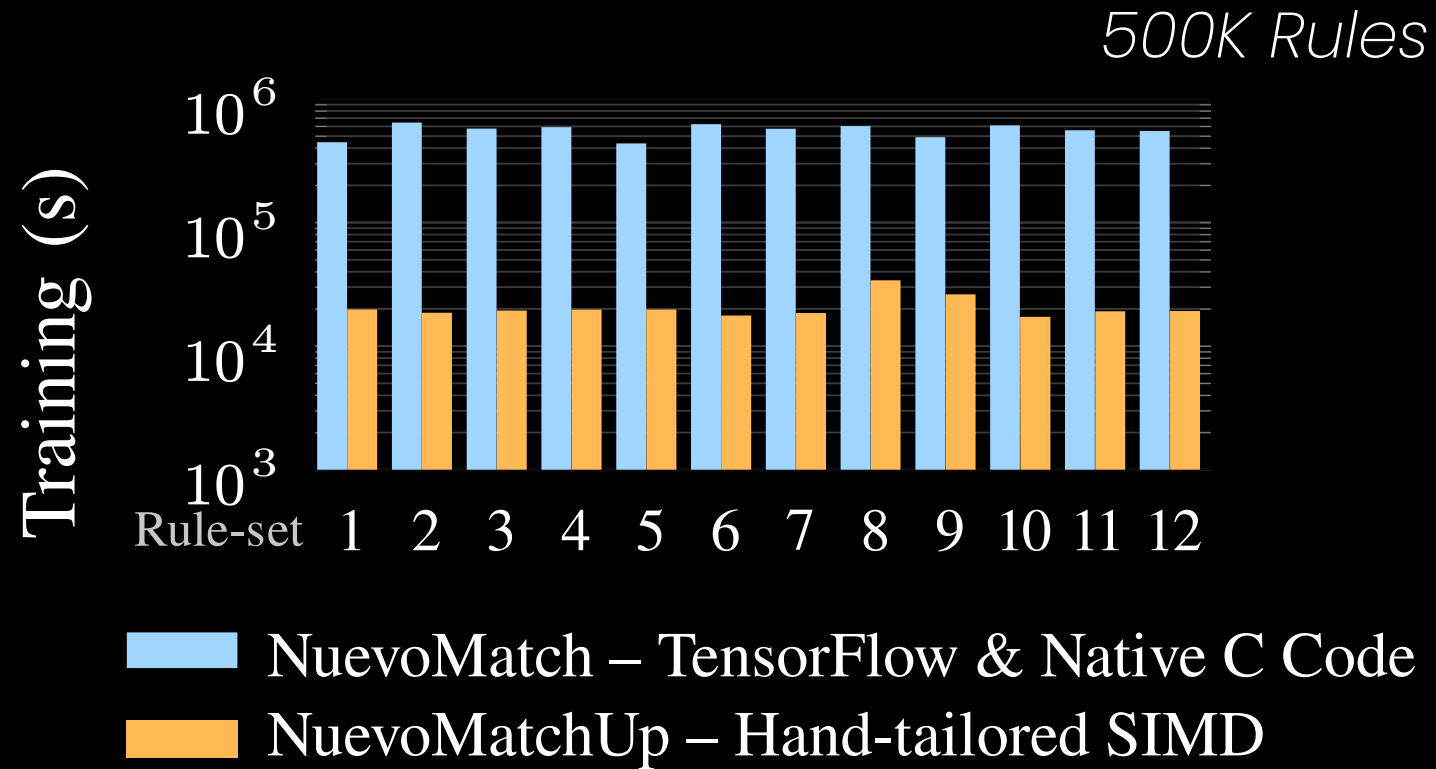
Efficient SIMD Implementation



introducing *NuevoMatch**UP***

speeding-up NuevoMatch updates

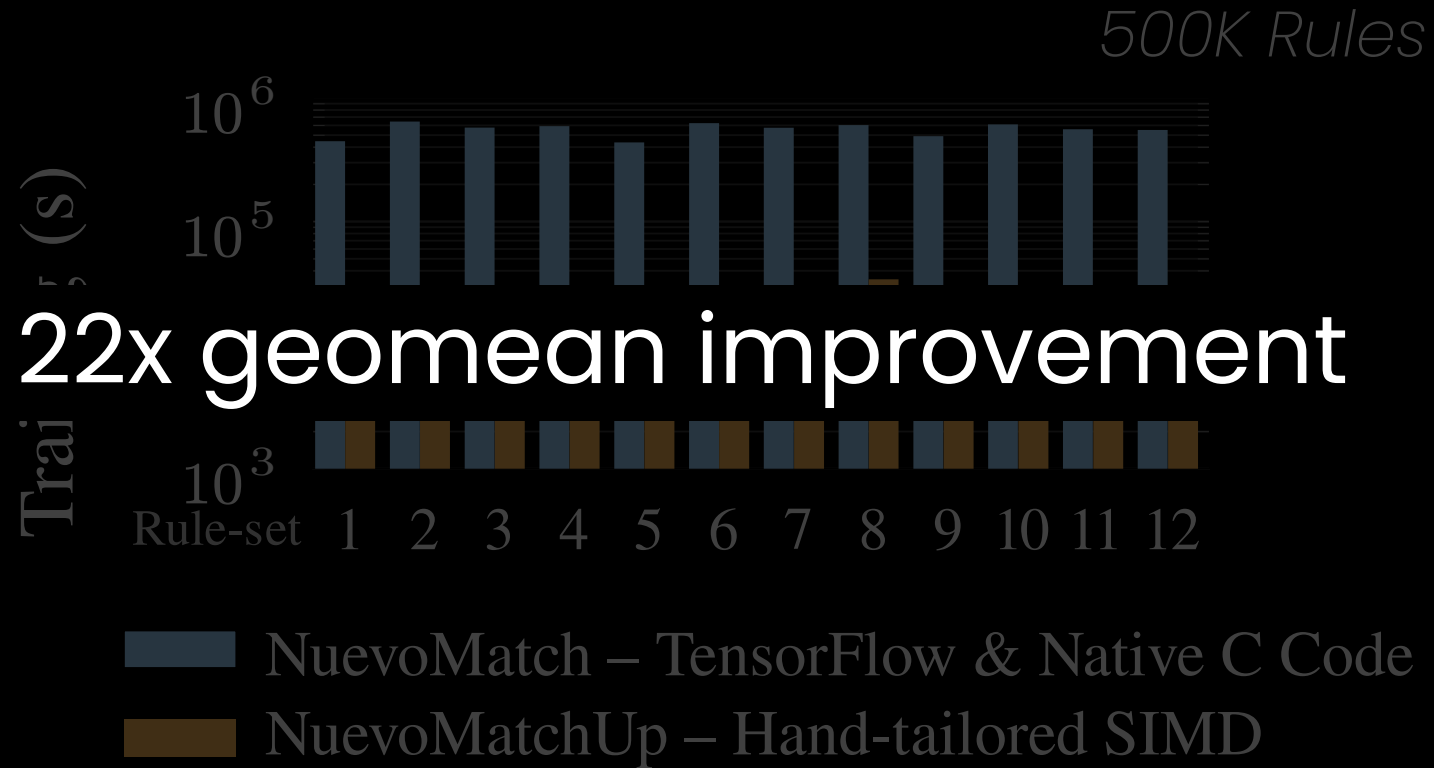
Efficient SIMD Implementation



introducing *NuevoMatch**UP***

speeding-up NuevoMatch updates

Efficient SIMD Implementation



22x geomean improvement

introducing *NuevoMatch**UP***

speeding-up NuevoMatch updates

Together:

Three orders of magnitude faster training

over NuevoMatch

introducing *NuevoMatch**UP***

speeding-up NuevoMatch updates

Together:

Three orders of magnitude faster training

over NuevoMatch

*NuevoMatch**UP*** can be integrated into OVS data-path!

Background

Challenge

This paper

Integration with OVS

Evaluation

*NuevoMatch**UP*** → OvS Open vSwitch

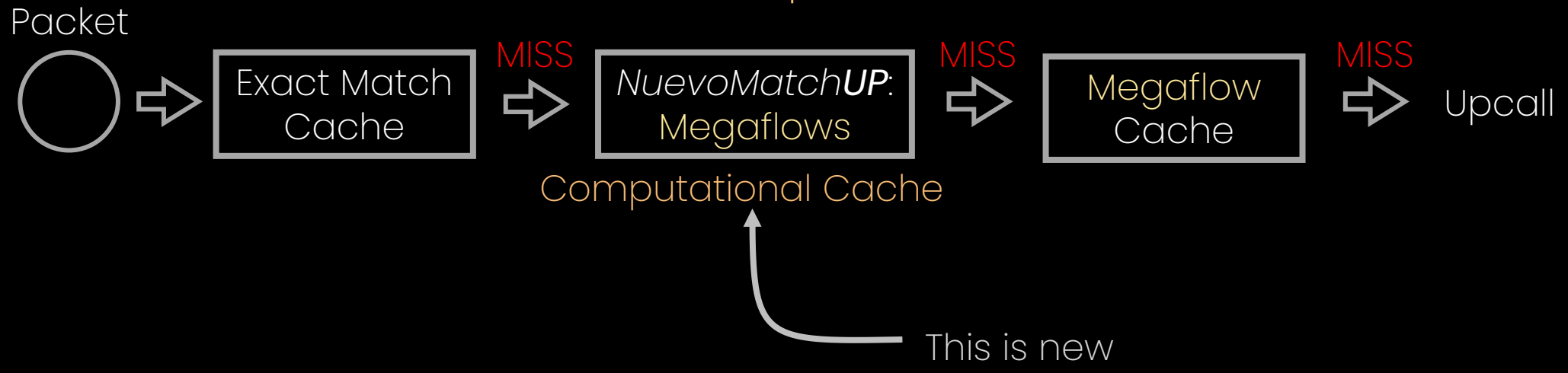
Two Design Options

for OVS data-path

*NuevoMatch**UP*** → OvS Open vSwitch

Two Design Options

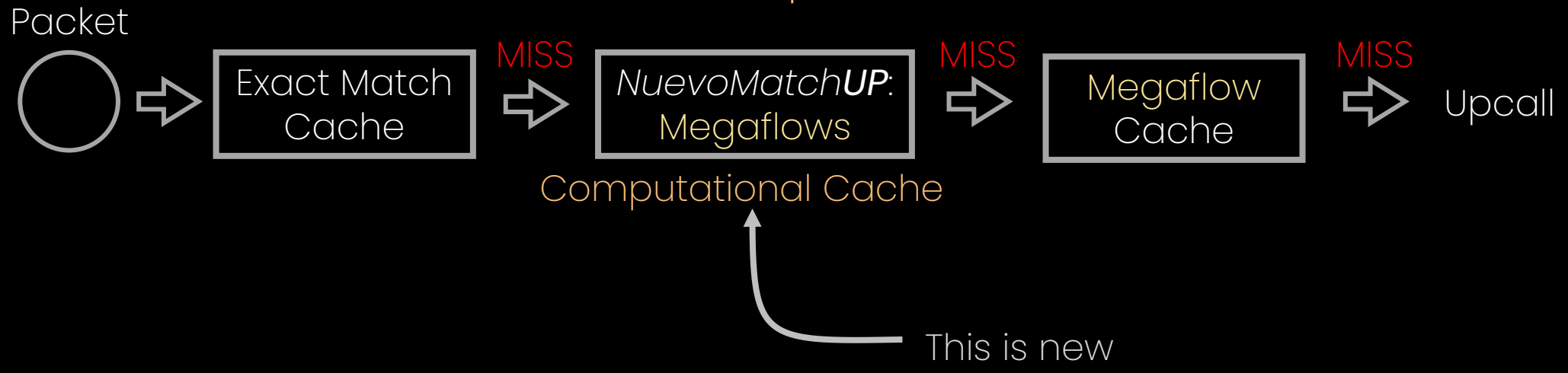
(1) OVS with a Computational Cache



*NuevoMatch**UP*** → OvS Open vSwitch

Two Design Options

(1) OVS with a Computational Cache

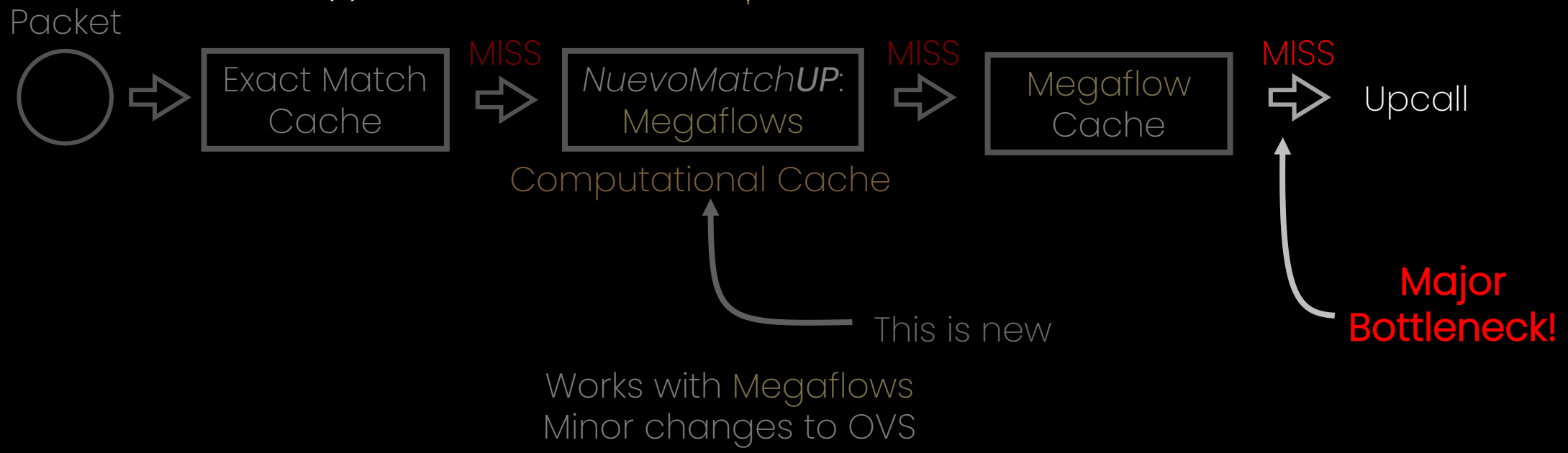


Works with Megaflows
Minor changes to OVS

*NuevoMatch**UP*** → OvS Open vSwitch

Two Design Options

(1) OVS with a Computational Cache

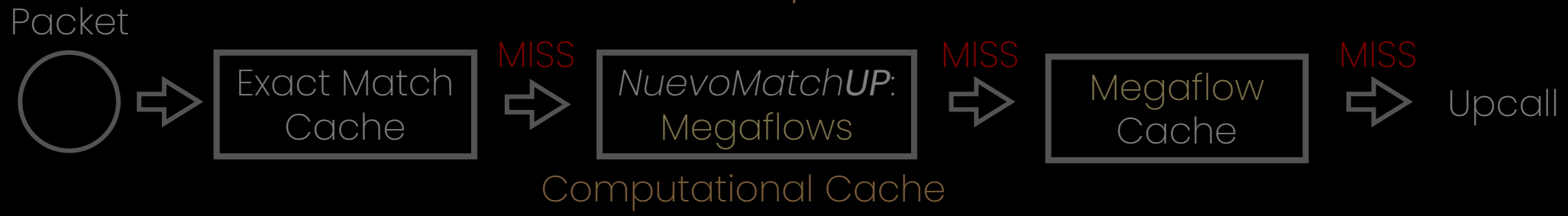


Works with Megaflows
Minor changes to OVS

*NuevoMatch**UP*** → OvS Open vSwitch

Two Design Options

(1) OVS with a Computational Cache

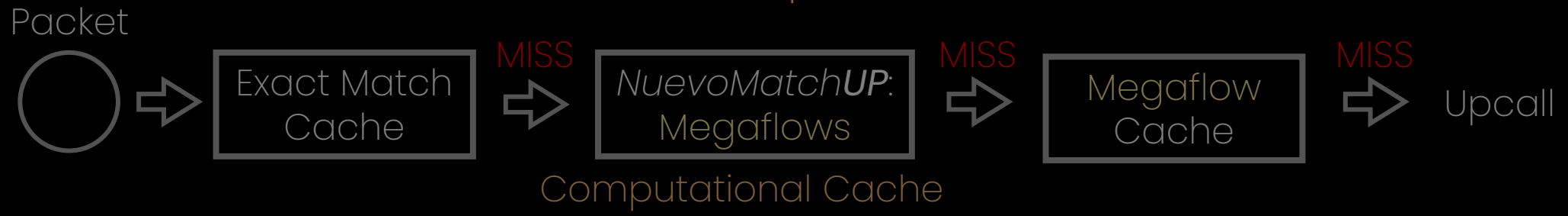


(2) OVS with Computational *Flows*

*NuevoMatch**UP*** → OvS Open vSwitch

Two Design Options

(1) OVS with a Computational Cache



(2) OVS with Computational *Flows*

A paradigm shift: data-path matching on OpenFlow rules

No Upcalls!

*NuevoMatch**UP*** → OvS Open vSwitch

Two Design Options

(1) OVS with a Computational Cache

We design, implement, and integrate both into OVS DPDK data-path

(2) OVS with Computational Flows

A paradigm shift: data-path matching on OpenFlow rules

No Upcalls!

Background

Challenge

This paper

Integration with OVS

Evaluation

*NuevoMatch**UP*** → OvS Open vSwitch

End to End Results

10Gbps NICs

ClassBench

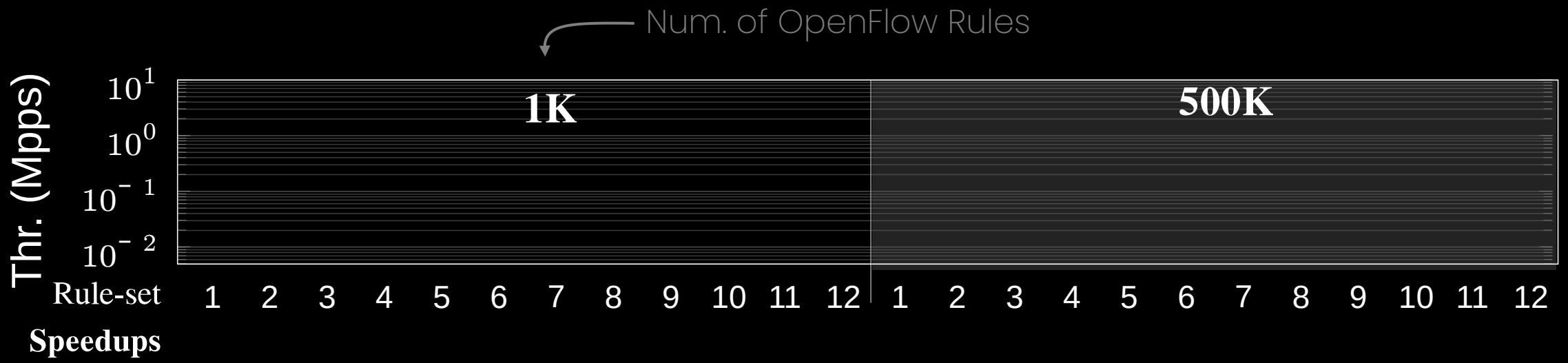
CAIDA

Max 1% Drops

NuevoMatchUP → OvS

Open vSwitch

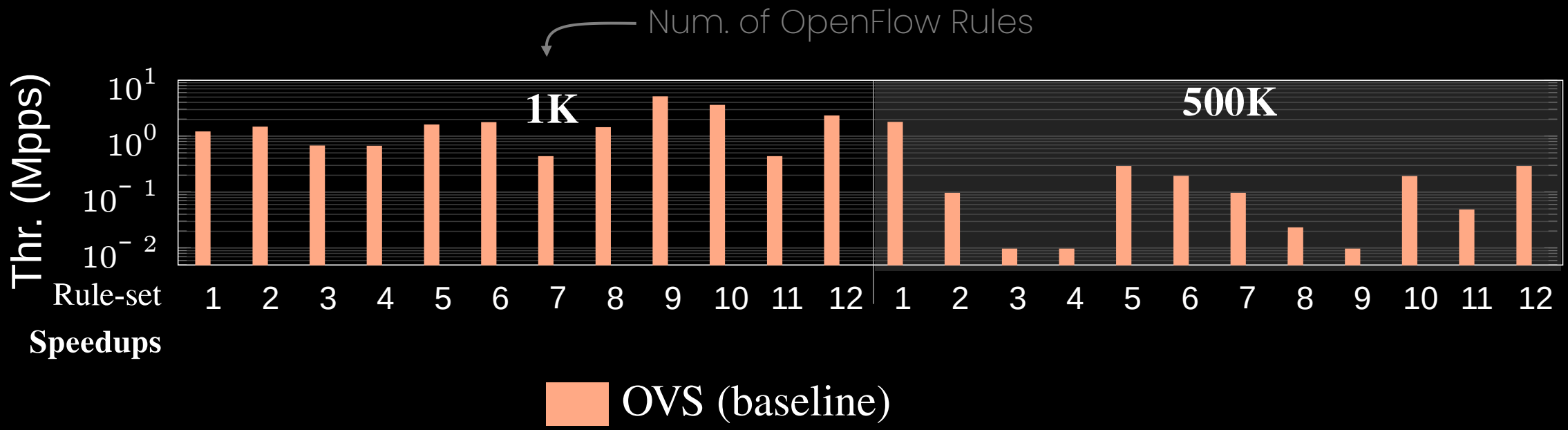
End to End Results



NuevoMatchUP → OvS

Open vSwitch

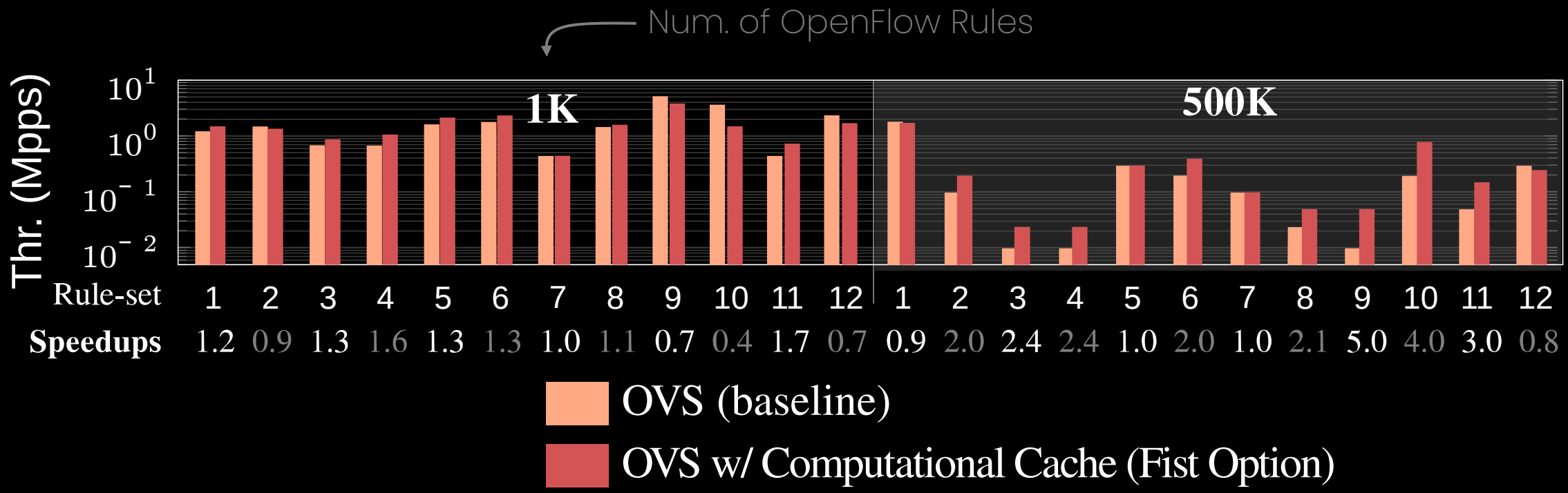
End to End Results



NuevoMatchUP → OvS

Open vSwitch

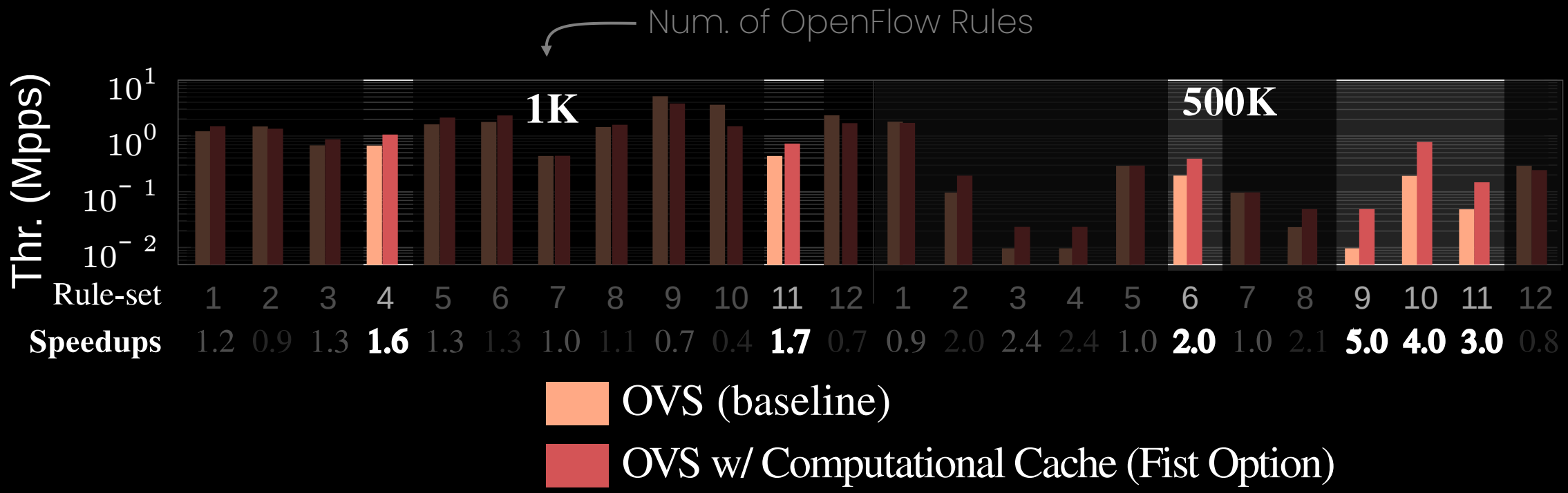
End to End Results



NuevoMatchUP → OvS

Open vSwitch

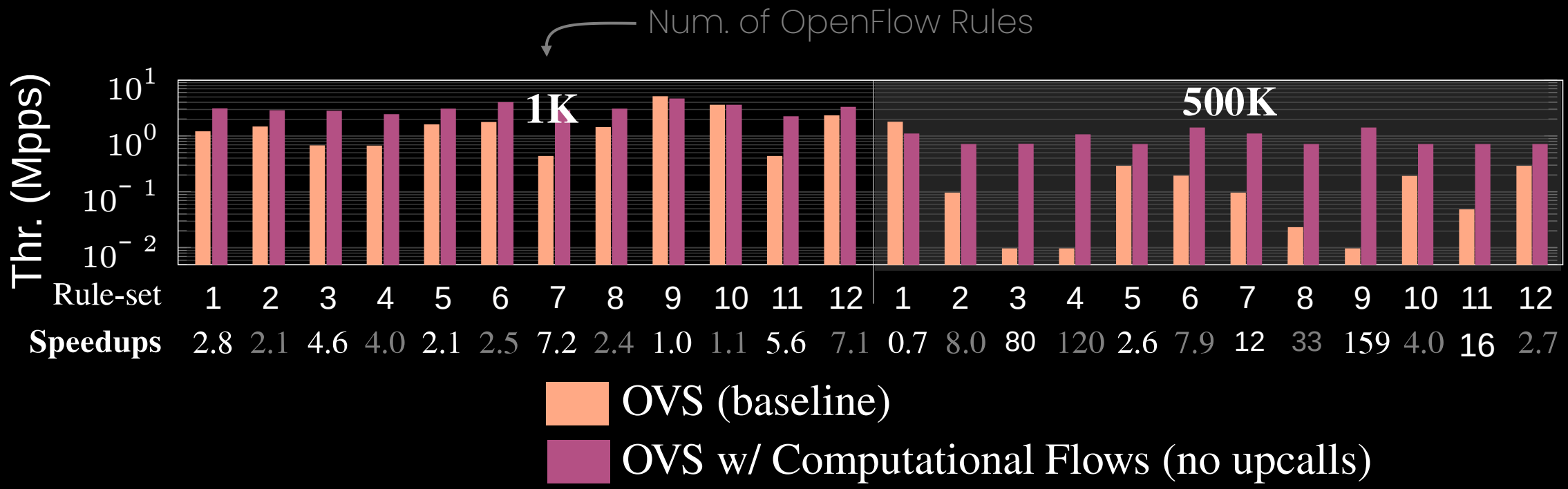
End to End Results



NuevoMatchUP → OvS

Open vSwitch

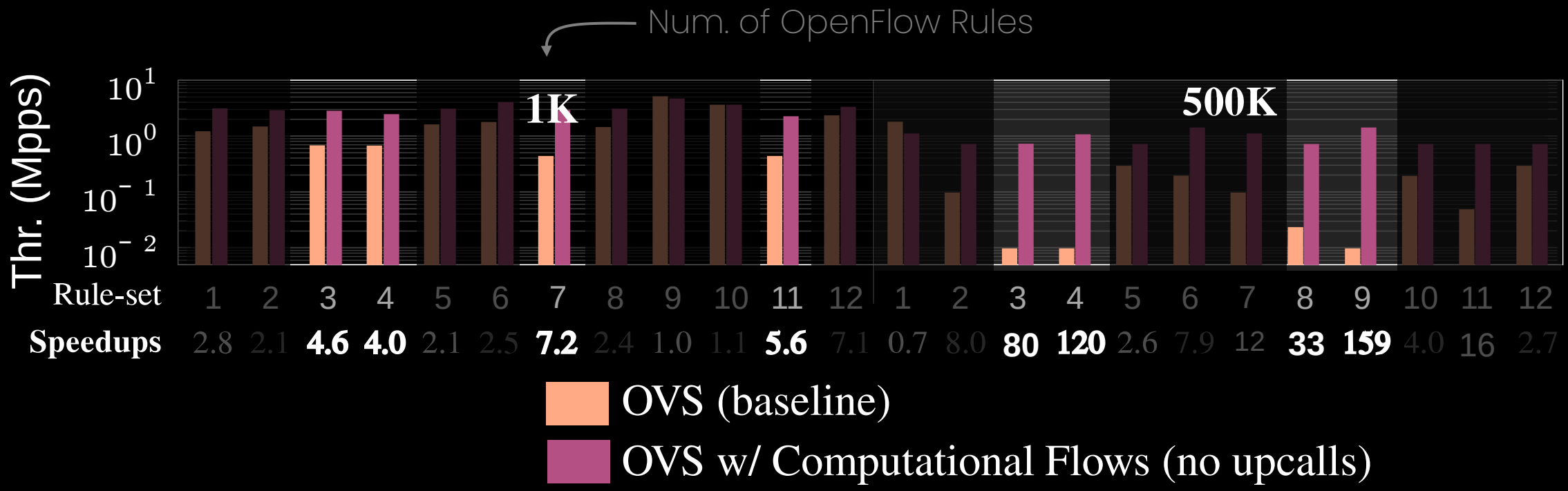
End to End Results



NuevoMatchUP → OvS

Open vSwitch

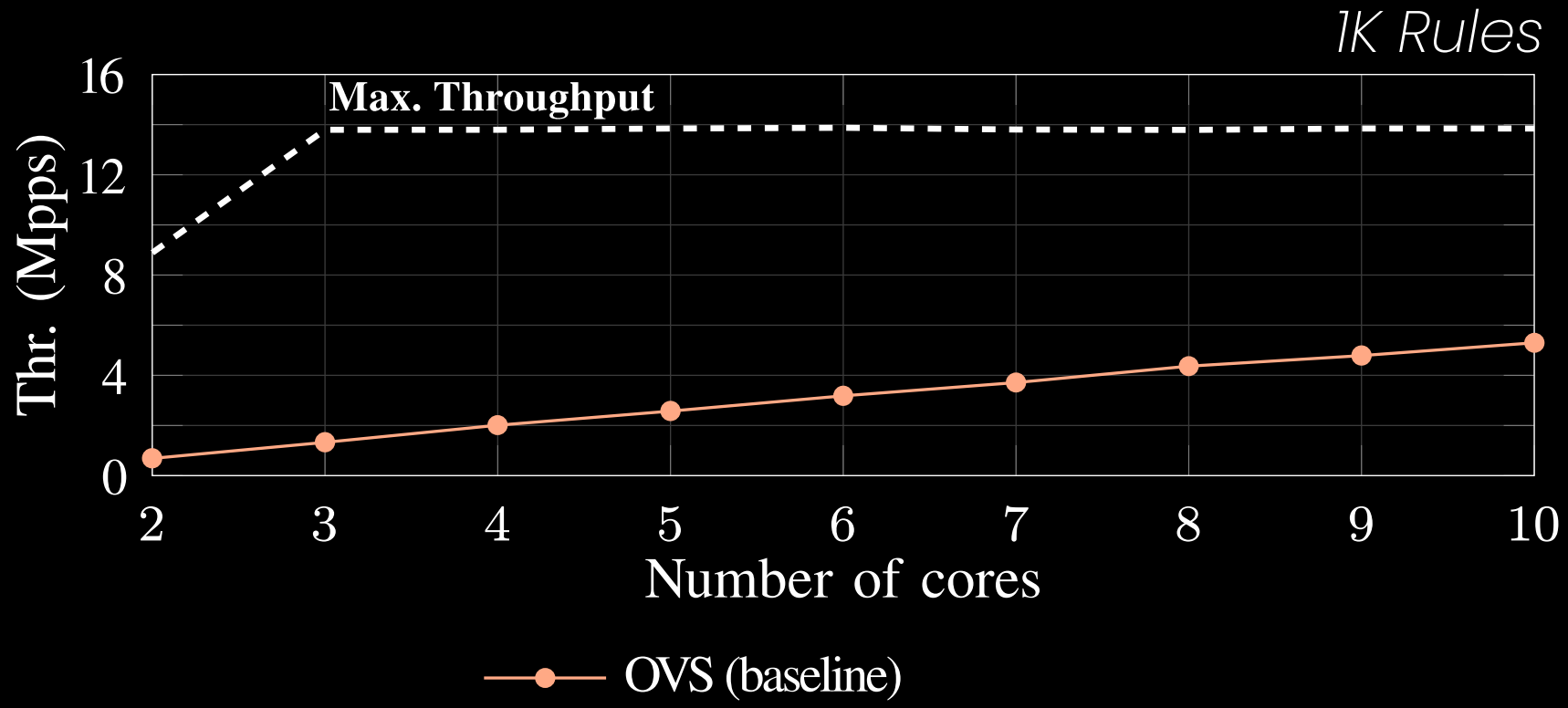
End to End Results



NuevoMatchUP → OvS

Open vSwitch

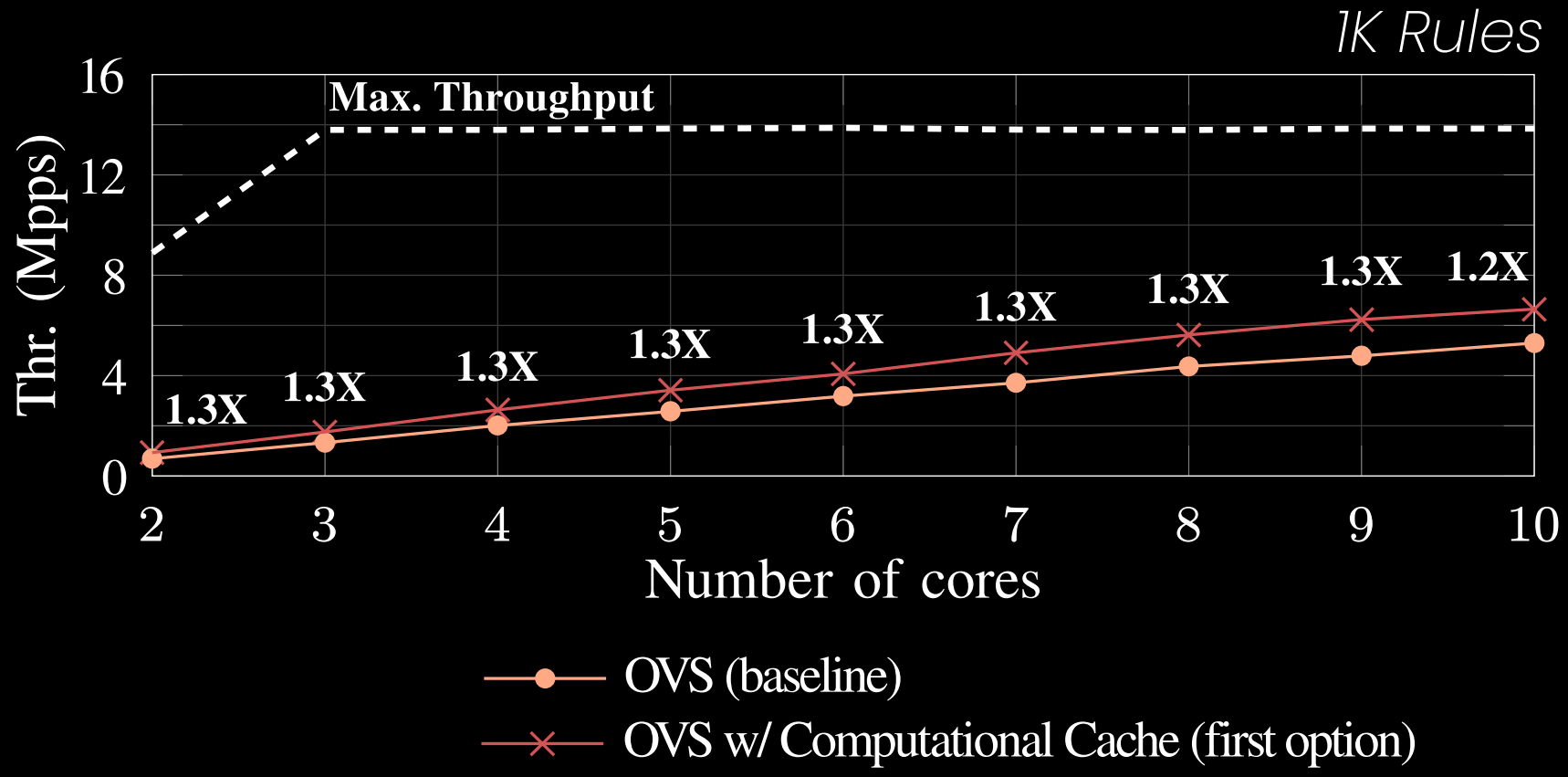
Scalable to #Cores



NuevoMatchUP → OvS

Open vSwitch

Scalable to #Cores

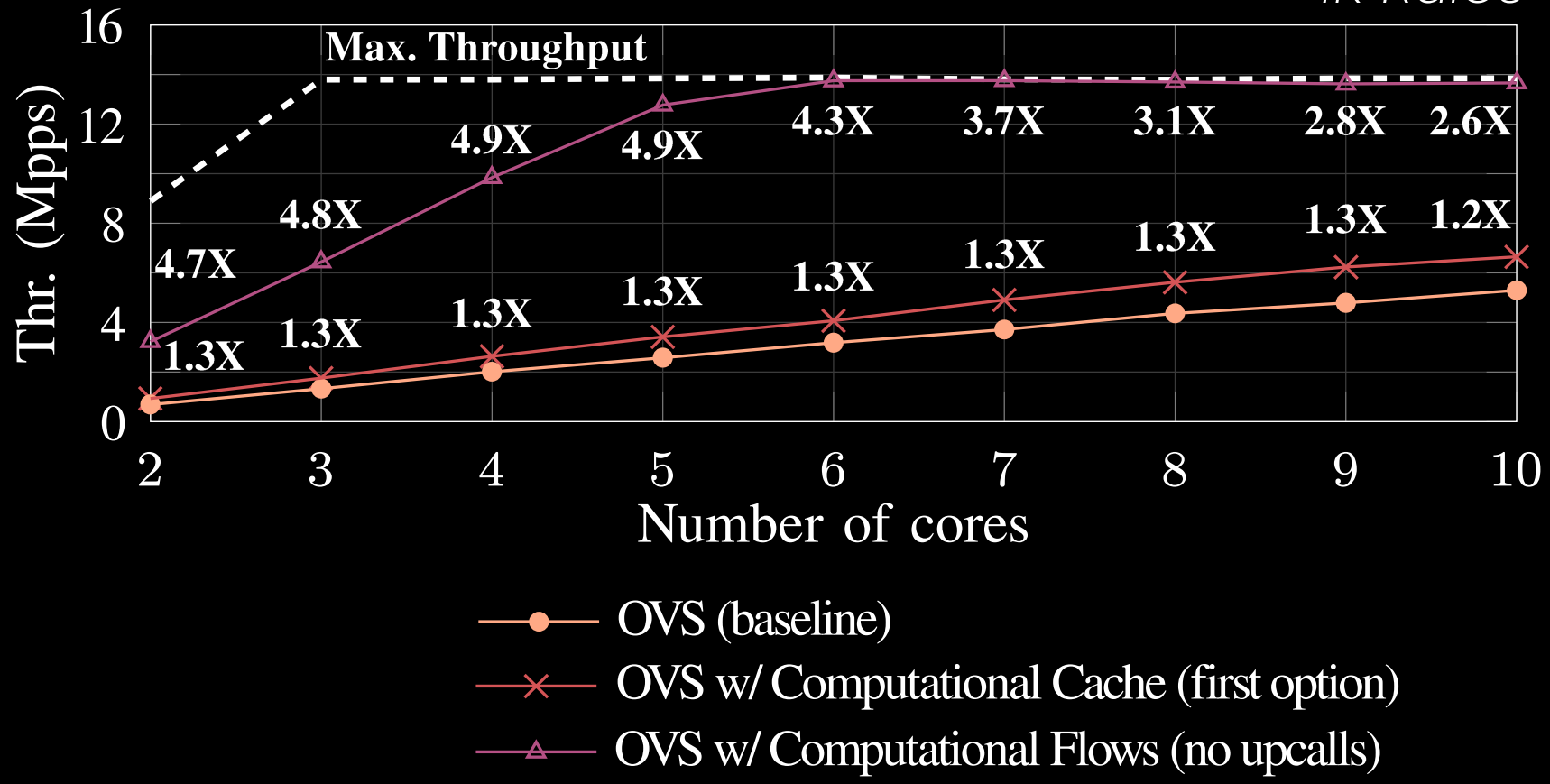


NuevoMatchUP → OvS

Open vSwitch

Scalable to #Cores

1K Rules



See Paper for

1. Throughput over time experiments
2. Throughput in the presence of updates
3. Microbenchmarks
4. And more...

Conclusions

1. Fast neural-net training for high update rates
2. New design of the OVS data-path
3. Readily deployable to commodity hardware

Future Work

1. NuevoMatchUP for P4
2. Reducing the need for training
3. Hardware acceleration

Future Work

1. NuevoMatchUP for P4
2. Reducing the need for training
3. Hardware acceleration

Thank You

Alon Rashelbach
alonrs@campus.technion.ac.il

Ori Rottenstreich
or@technion.ac.il

Mark Silberstein
mark@ee.technion.ac.il