Taurus: A Data Plane Architecture for Per-Packet ML

Tushar Swamy

Alexander Rucker, Muhammad Shahbaz, Ishan Gaur, and Kunle Olukotun

Stanford University

Datacenter networks are becoming harder to manage…

> “ *Our current generation — Jupiter fabrics — can deliver more than 1 Petabit/sec of total bisection bandwidth* ”

— A Look Inside Google’s Data Center Networks[1]

Networks require complex management with high performance

Automate decision-making with machine learning (ML)

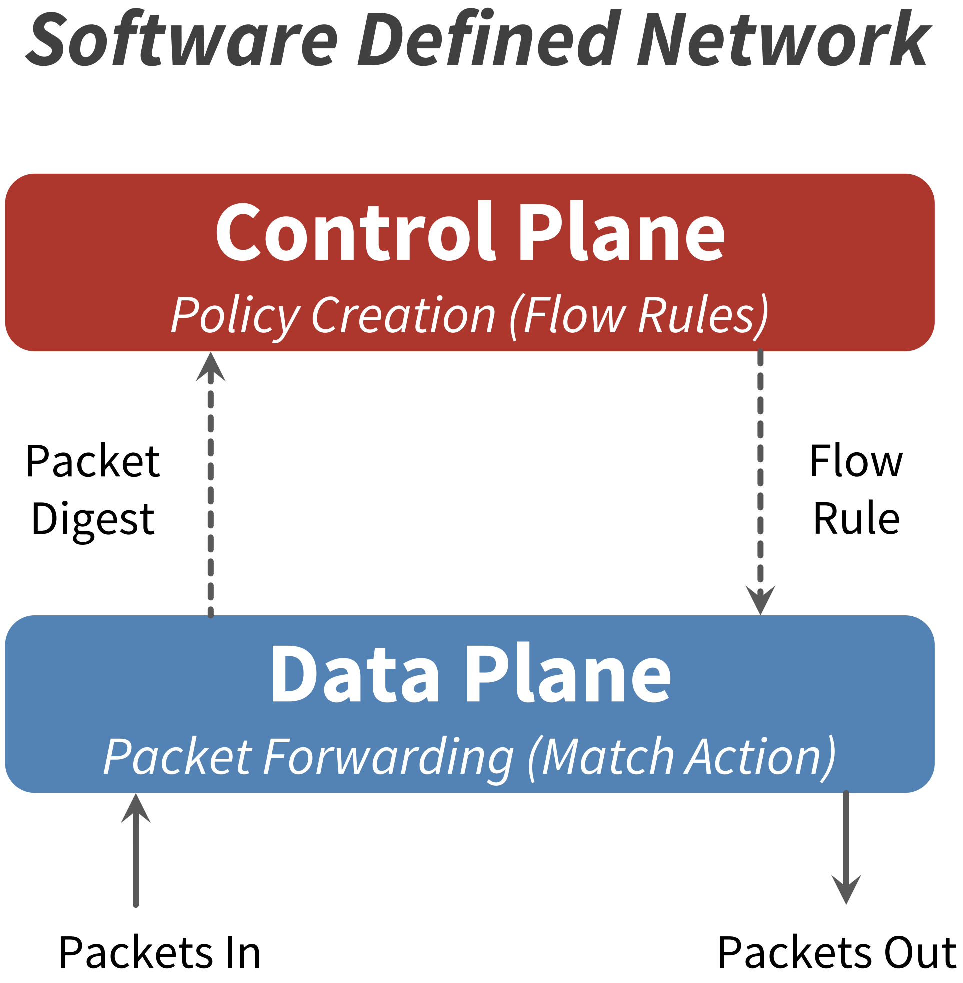
- Making decisions based on data → ***machine learning***

- Machine learning can:

  - ***Approximate*** network functions based on data
  - ***Customize*** network functions based on data

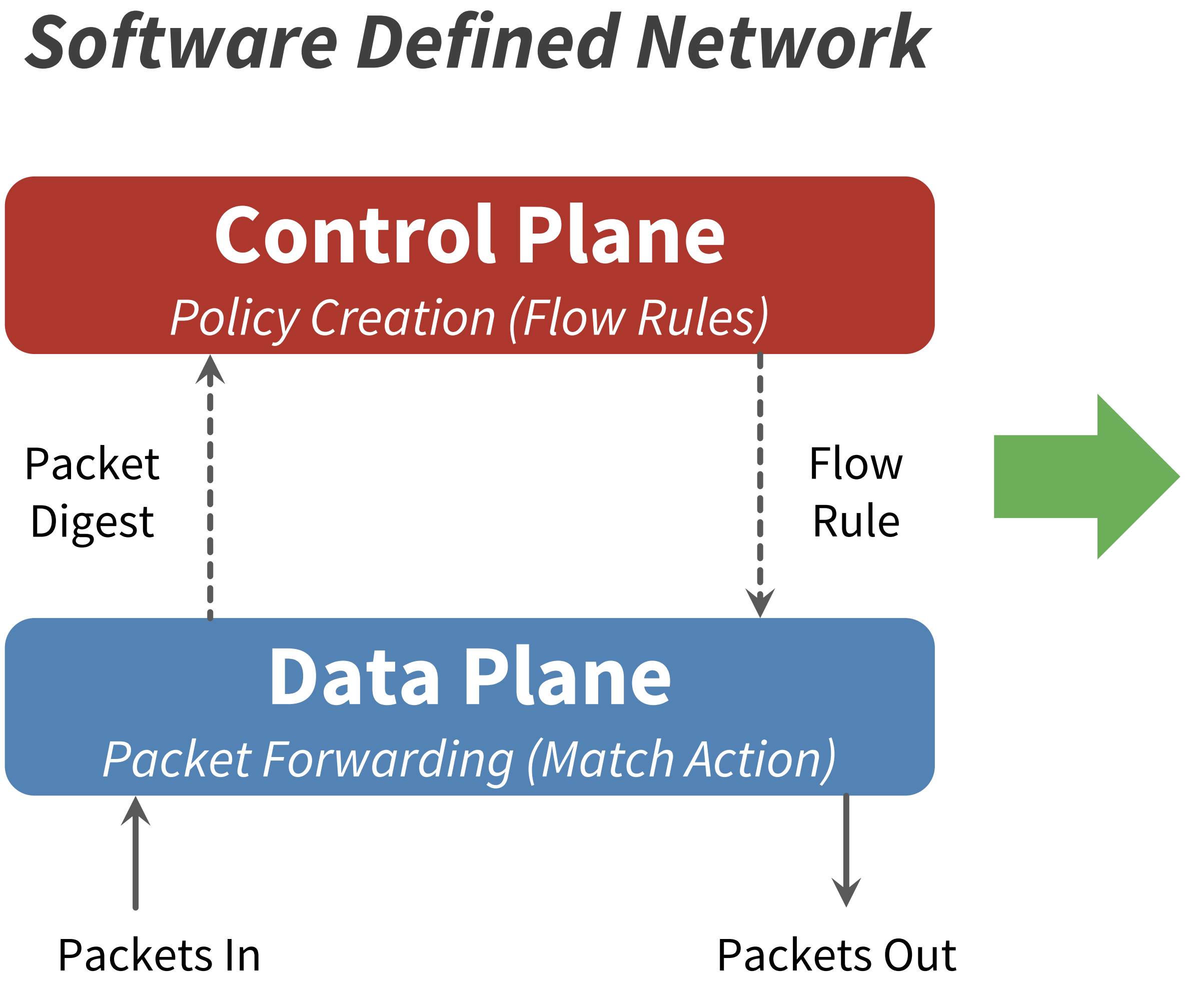
- Currently, we use by hand-written heuristics in the network…

Where in the network should ML happen?

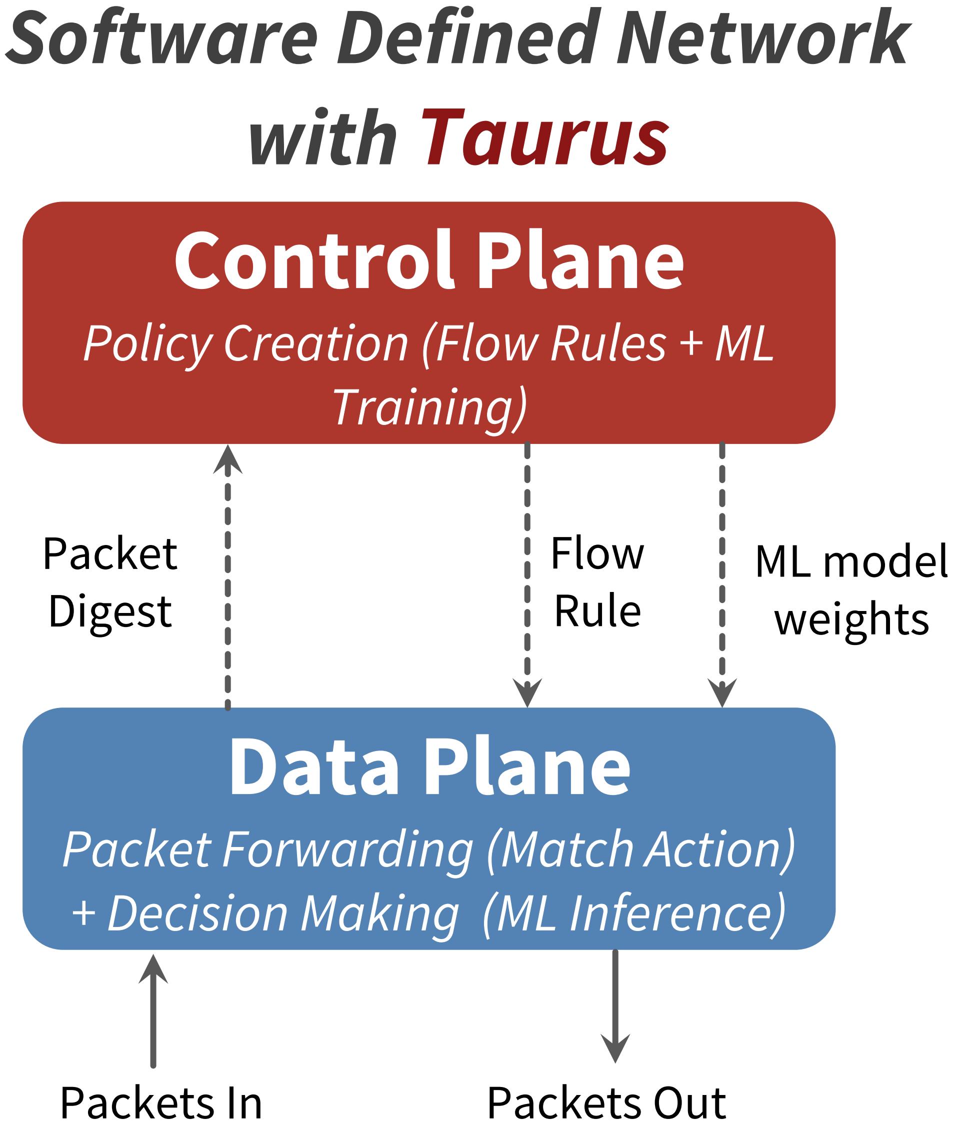


Software Defined Network
Control Plane
Policy Creation (Flow Rules)
Packet Digest
Flow Rule
Data Plane
Packet Forwarding (Match Action)
Packets In
Packets Out

A Taurus network introduces ML for management



Software Defined Network



Software Defined Network with Taurus

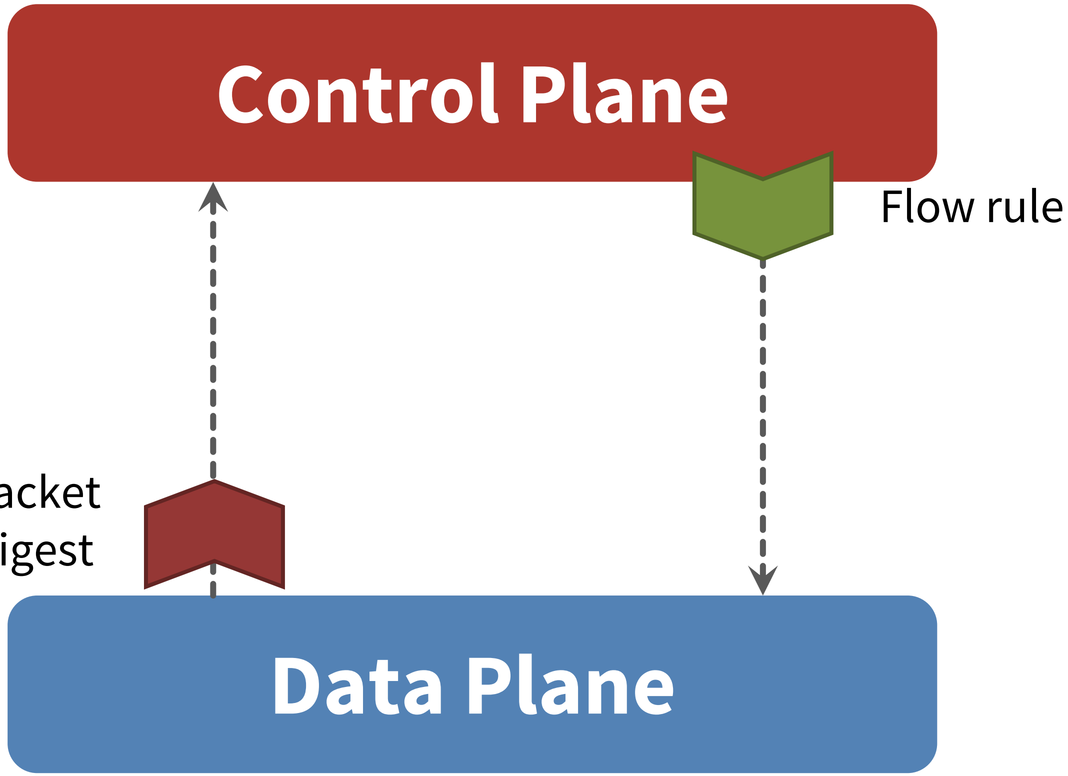
Control Plane
Policy Creation (Flow Rules + ML Training)
Packet Digest
Flow Rule
ML model weights
Data Plane
Packet Forwarding (Match Action) + Decision Making (ML Inference)
Packets In
Packets Out

ML inference should happen *per-packet* in the *data plane*

Example: Anomaly Detection

Processing time: [illegible]

Packets missed: [illegible]



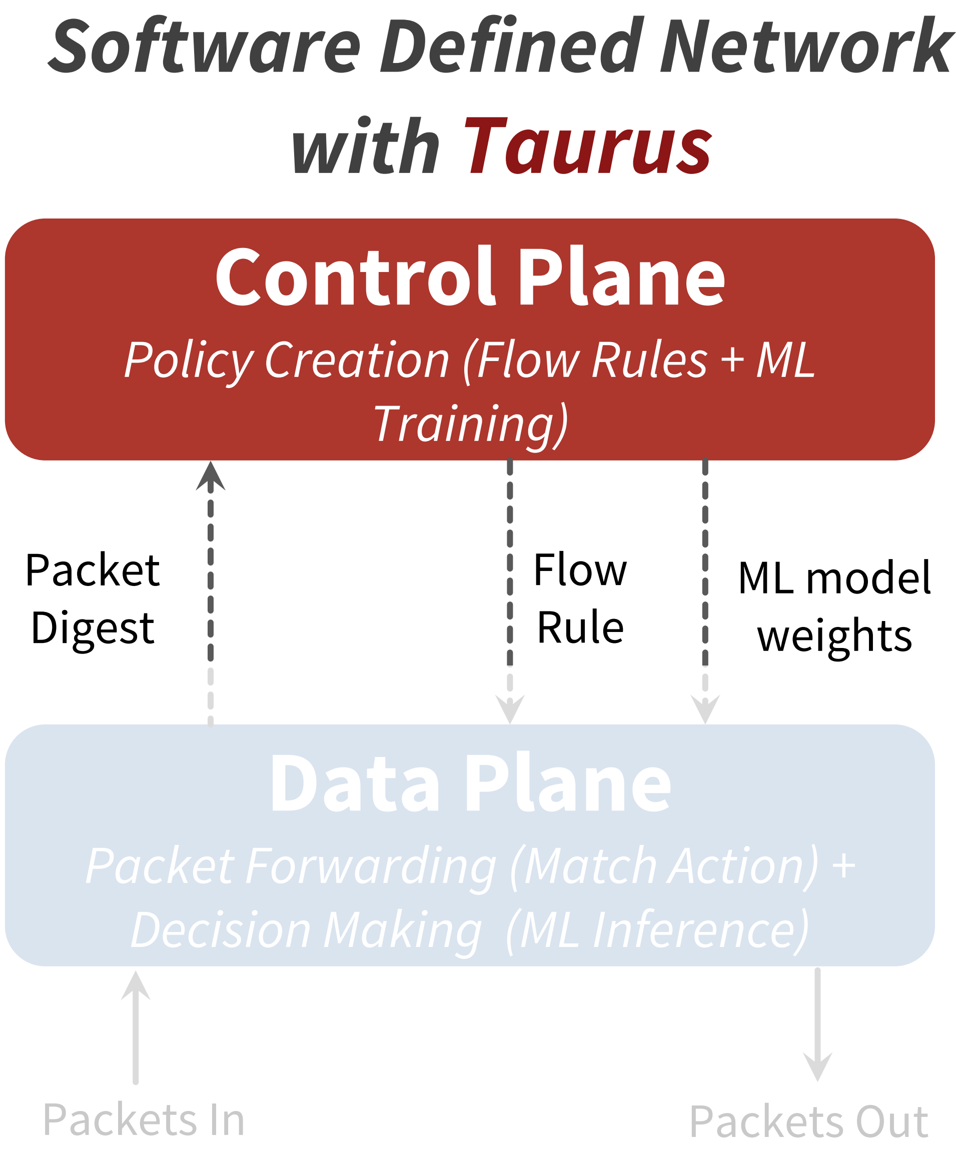
1.5 M Packets missed during flow rule installation time

Robustness and performance of the network are determined by:

- *Quality of reaction*
- *Speed of reaction*

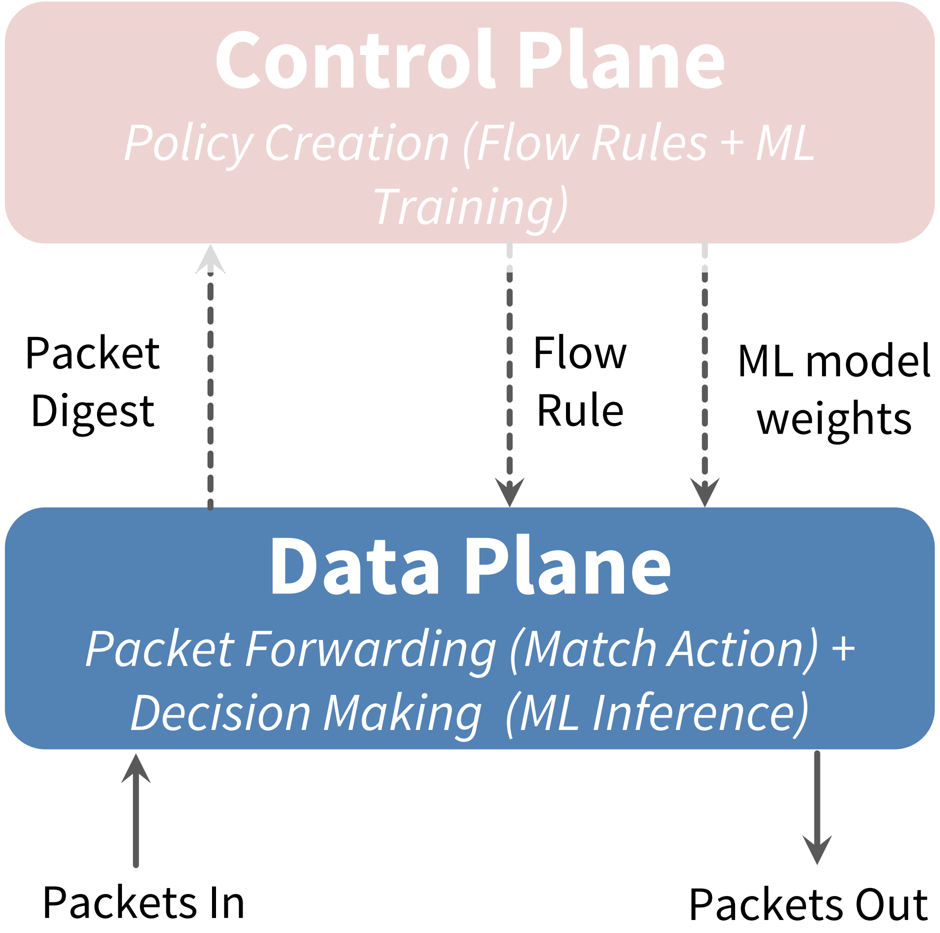
ML training happens in the control plane

ML Training is off critical path



ML Inference happens in the data plane

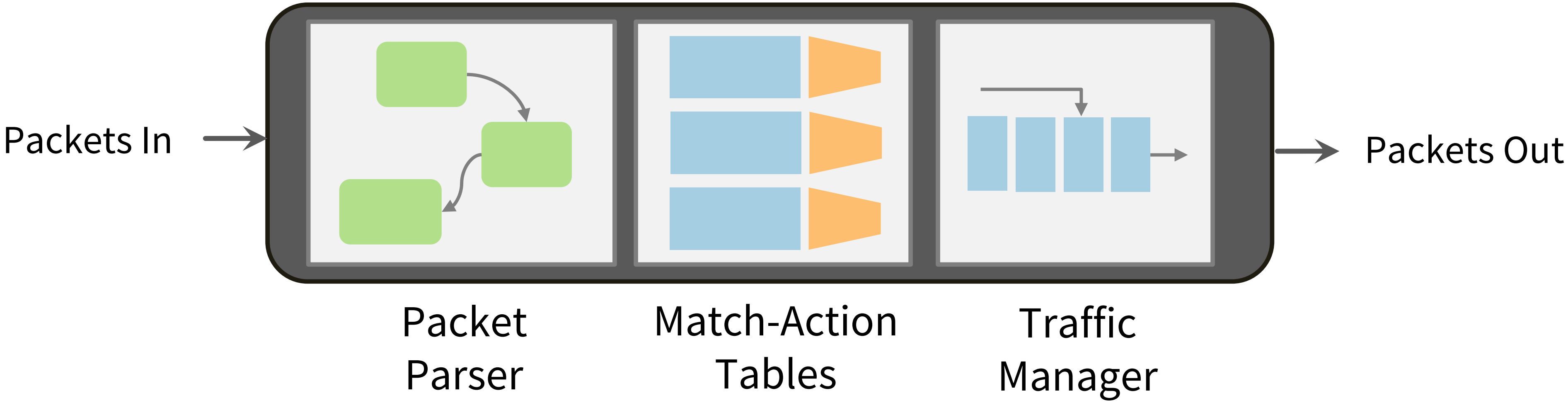
Software Defined Network with Taurus


Control Plane
Policy Creation (Flow Rules + ML Training)
Packet Digest
Flow Rule
ML model weights
Data Plane
Packet Forwarding (Match Action) + Decision Making (ML Inference)
Packets In
Packets Out

ML Inference is on critical path

Taurus **is an architecture for per-packet ML inference in the data plane**

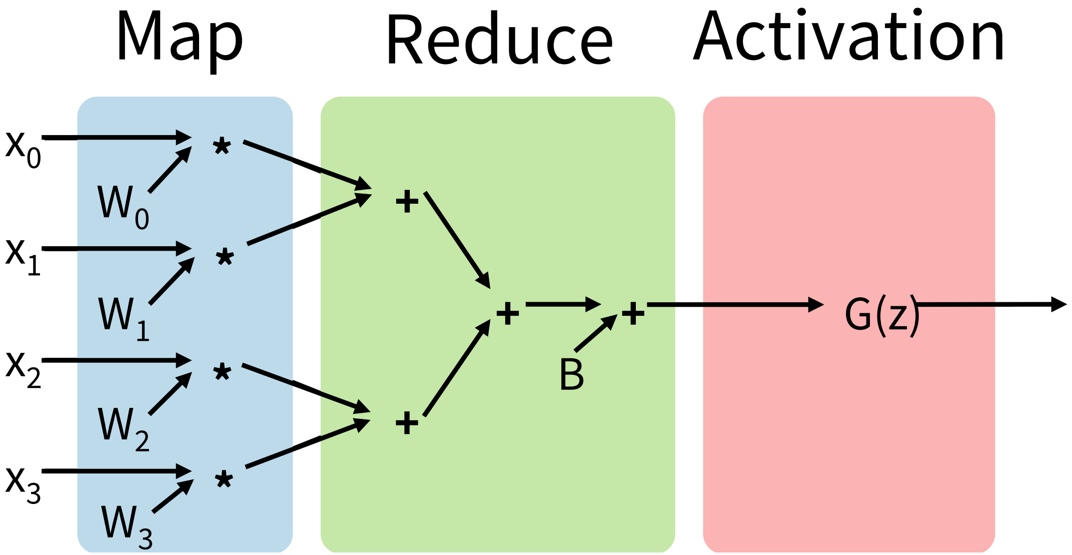
What do programmable switches look like?



A Protocol Independent Switch Architecture (PISA)

What abstraction should we use?

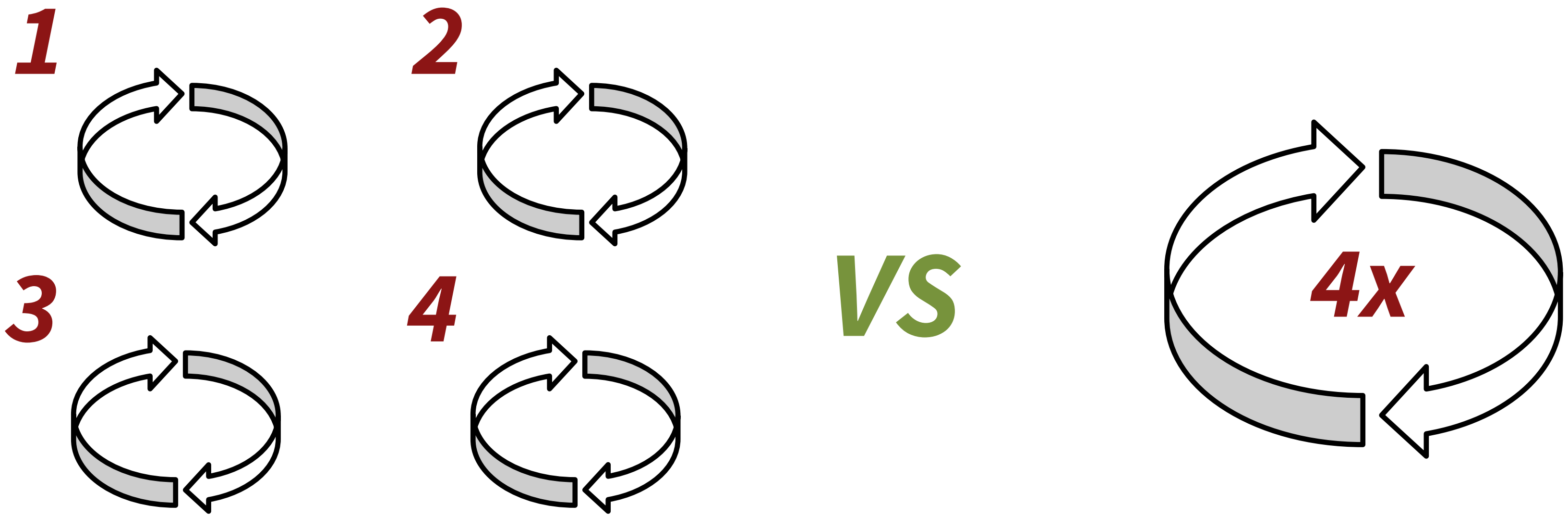
- ***Map-reduce*** can support linear algebra operations common in ML algorithms
  - Ex. Operations) Dot products, matrix multiplications, etc.
  - Ex. Algorithms) Neural networks, support vector machines



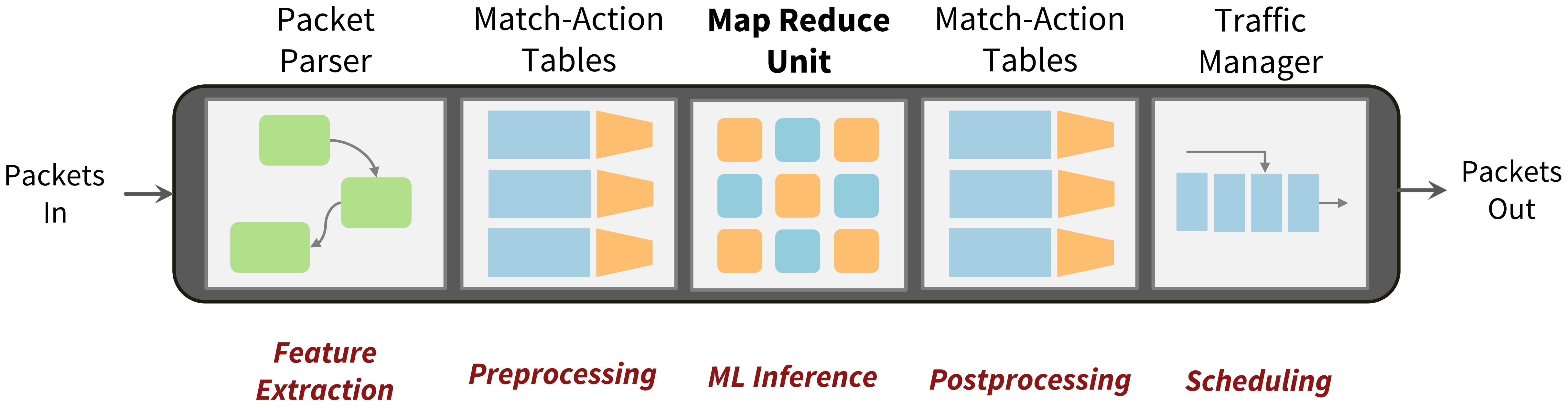
What abstraction should we use?

- ***SIMD Parallelism*** enables performance with minimal logic
  - VLIW pipelines require too much communication hardware (e.g Tofino)

- ***Unrolling*** patterns allows for flexibility
  - More unrolling → better performance
  - Less unrolling → less resource usage

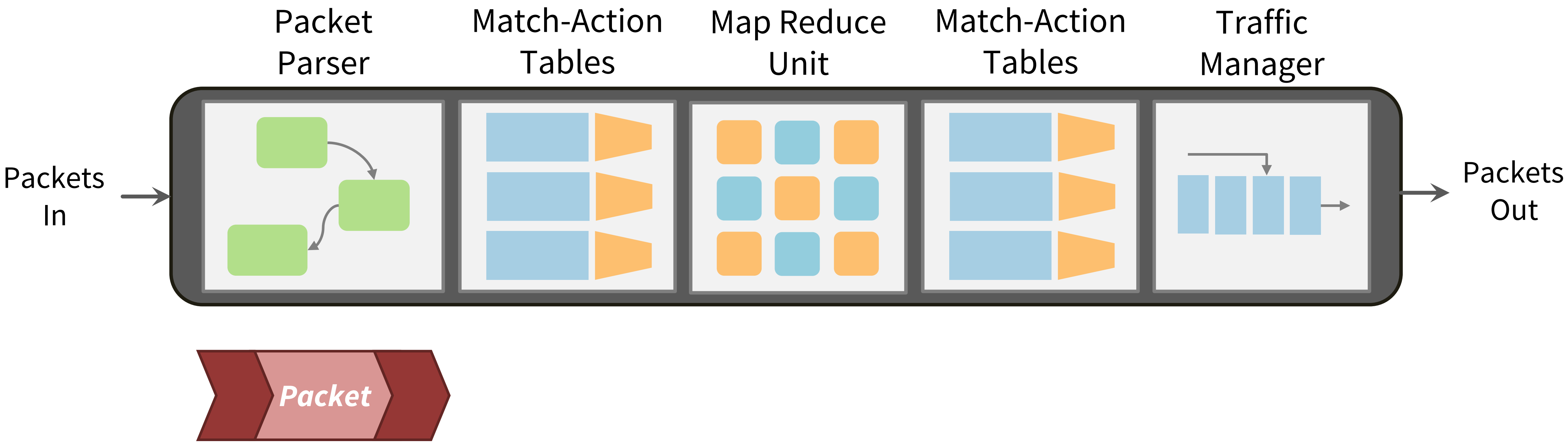


The Taurus pipeline with a Map Reduce Unit



- ***Map Reduce Unit*** must:
  - be reconfigurable
  - meet line rate (with a fixed clock)
  - incur minimal area and power overhead

Example Application: Anomaly Detection



Read local features (e.g., IP address)

Retrieve out of network events (e.g., failed logins per IP)

Apply learned anomaly detection

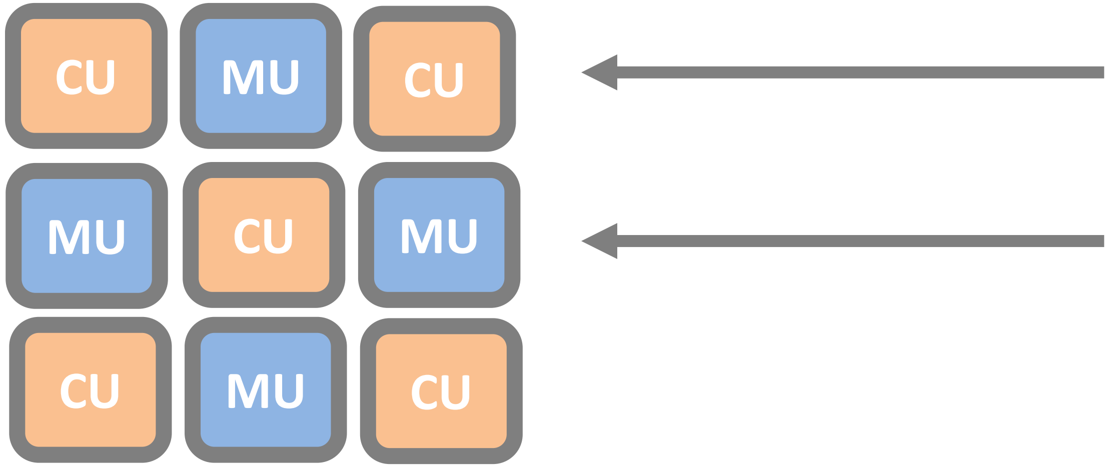
Select a port or action (e.g., drop if score == 1)

Send packet to destination

Evaluation of a Taurus ASIC

- Our evaluation platform is based on ***Plasticine***

- We program our map-reduce applications in the ***Spatial HDL***



CU MU CU
MU CU MU
CU MU CU

Compute unit

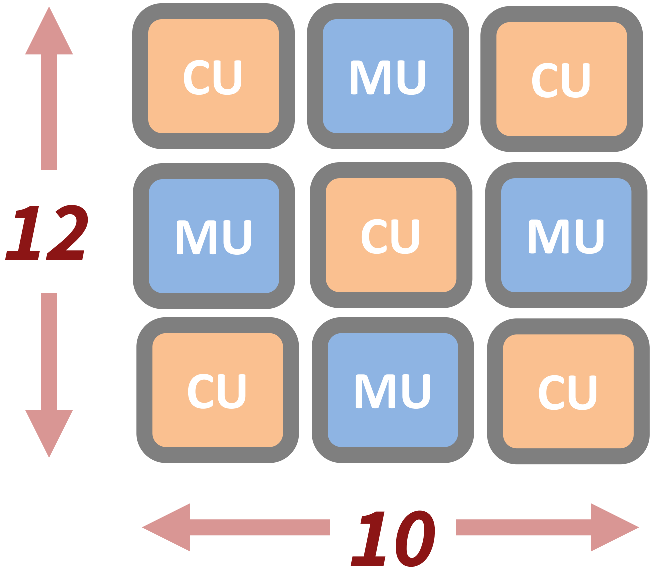
Memory unit

More architectural details in full paper!

Evaluation of a Taurus ASIC

- Our evaluation platform is based on ***Plasticine***

- We program our map-reduce applications in the ***Spatial HDL***



	Area	
Hardware	mm^2	+%
12x10 MR Grid	4.8 x 4	3.8
Prog. Switch	500	---

**Overheads are calculated relative to state of the art programmable switches*

Evaluation of an Anomaly Detection (AD) benchmark

- ***AD SVM: 8 support vectors***
- ***AD DNN: 4 layers - 12x6x3x2 neurons***

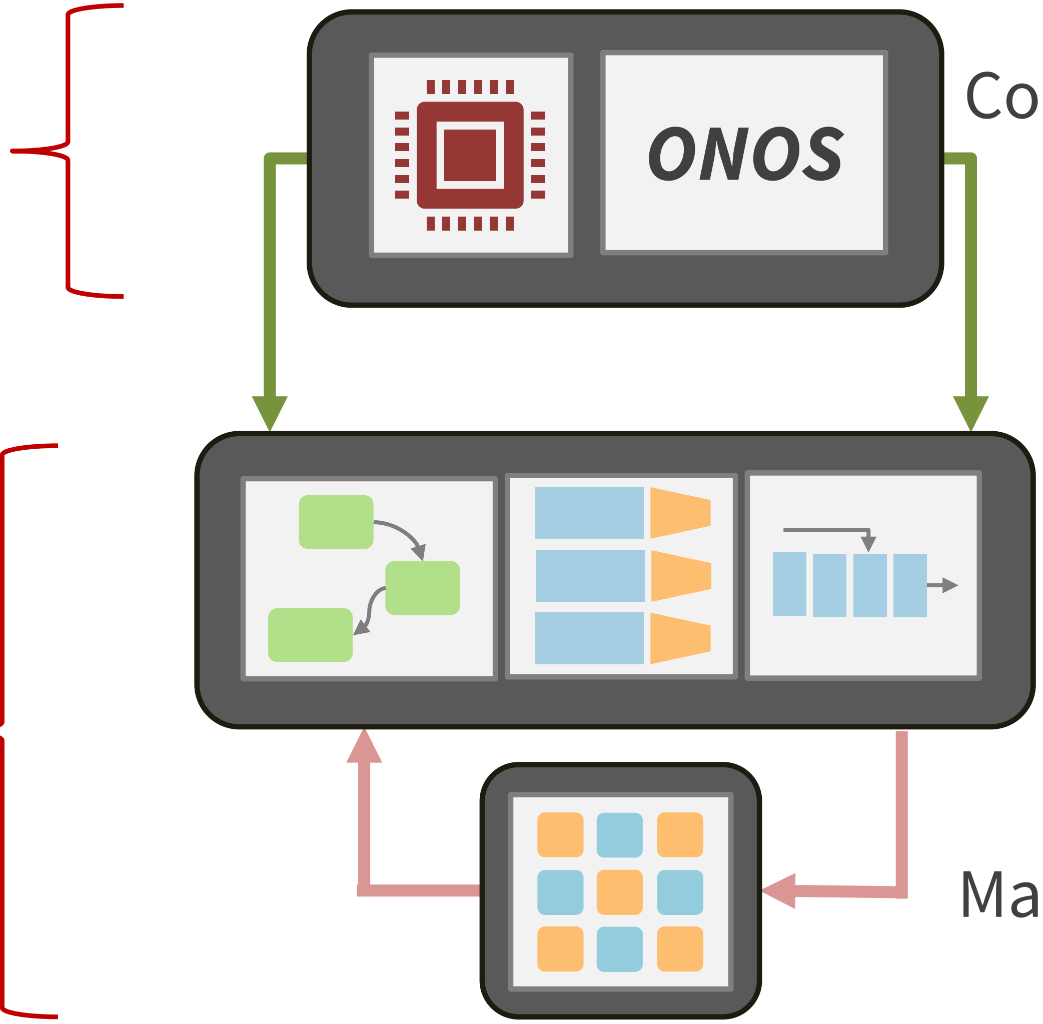
Overhead of Map Reduce Unit

			Area	Power
Model	TP (GPkt/s)	Lat (ns)	+%	+%
SVM	1	83	0.5	0.6
DNN	1	221	0.8	1.0

**Overheads are calculated relative to state of the art programmable switches*

More apps in full paper!

We provide an open-source, FPGA-based testbed



Control Plane

Controller: CPU

Data Plane

PISA pipeline: Tofino Switch

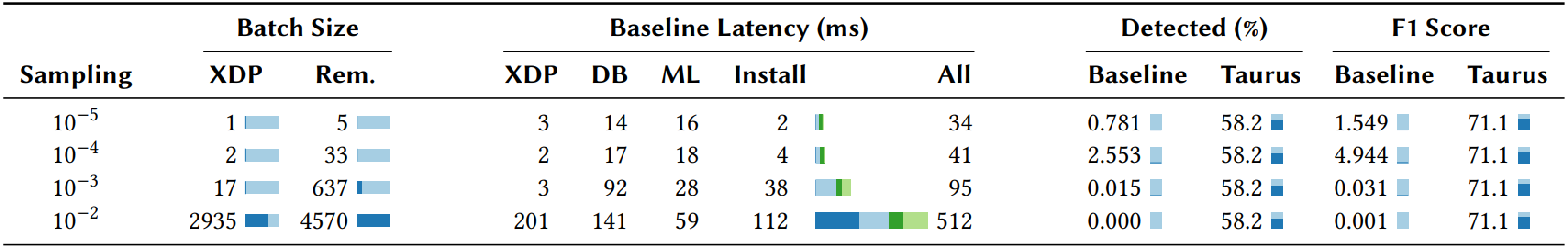
Map Reduce Unit: FPGA

FPGA-based testbed evaluations

- ***FPGA Testbed*** enables both control plane ML (baseline) and data plane ML (Taurus) evaluations

- ***ML anomaly detection*** is evaluated on both control plane and data plane

- ***Control plane latency*** directly affects the accuracy of the ML model, rendering it useless



	Batch Size		Baseline Latency (ms)					Detected (%)		F1 Score	
Sampling	**XDP**	**Rem.**	**XDP**	**DB**	**ML**	**Install**	**All**	**Baseline**	**Taurus**	**Baseline**	**Taurus**
10^{-5}	1	5	3	14	16	2	34	0.781	58.2	1.549	71.1
10^{-4}	2	33	2	17	18	4	41	2.553	58.2	4.944	71.1
10^{-3}	17	637	3	92	28	38	95	0.015	58.2	0.031	71.1
10^{-2}	2935	4570	201	141	59	112	512	0.000	58.2	0.001	71.1

Questions?

Tushar Swamy

tswamy@stanford.edu

Read the paper:

https://dl.acm.org/doi/10.1145/3503222.3507726

Try it out!

https://gitlab.com/dataplane-ai/taurus