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

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**

### Control Plane
- Policy Creation (Flow Rules)

### Data Plane
- Packet Forwarding (Match Action)

### Packet Digest
- Packets In
- Packets Out

---

**Software Defined Network with Taurus**

### Control Plane
- Policy Creation (Flow Rules + ML Training)

### Data Plane
- Packet Forwarding (Match Action) + Decision Making (ML Inference)

### ML model
-weights

### Packet Digest
- Packets In
- Packets Out

---

3

Packets In

Packets Out
ML inference should happen *per-packet* in the *data plane*
Example: Anomaly Detection

Processing time: 0.5 ms
Packets missed: 1.5M

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

**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 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)

Data Plane
Packet Forwarding (Match Action) + Decision Making (ML Inference)

ML Inference is on critical path

Packet Digest → Flow Rule → ML model weights
Packets In → Packets Out
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

1 2 3 4

VS

4x
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

Packet Parser

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

Match-Action Tables

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

Map Reduce Unit

Apply learned anomaly detection

Match-Action Tables

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

Traffic Manager

Send packet to destination

Packets In

Packets Out
Evaluation of a Taurus ASIC

- Our evaluation platform is based on *Plasticine*

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

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*

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Area</th>
<th>+%</th>
</tr>
</thead>
<tbody>
<tr>
<td>12x10 MR Grid</td>
<td>4.8 x 4</td>
<td>3.8</td>
</tr>
<tr>
<td>Prog. Switch</td>
<td>500</td>
<td>---</td>
</tr>
</tbody>
</table>

*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

<table>
<thead>
<tr>
<th>Model</th>
<th>TP (GPkt/s)</th>
<th>Lat (ns)</th>
<th>Area +%</th>
<th>Power +%</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>1</td>
<td>83</td>
<td>0.5</td>
<td>0.6</td>
</tr>
<tr>
<td>DNN</td>
<td>1</td>
<td>221</td>
<td>0.8</td>
<td>1.0</td>
</tr>
</tbody>
</table>

*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
  - PISA pipeline: Tofino Switch

- **Data Plane**
  - Map Reduce Unit: FPGA

**Diagram Components:**
- ONOS
- 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

<table>
<thead>
<tr>
<th>Sampling</th>
<th>Batch Size</th>
<th>Baseline Latency (ms)</th>
<th>Detected (%)</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>XDP  DB  ML  Install  All</td>
<td>Baseline  Taurus</td>
<td>Baseline  Taurus</td>
</tr>
<tr>
<td>10^{-5}</td>
<td>1</td>
<td>3  14  16  2  34</td>
<td>0.781  58.2</td>
<td>1.549  71.1</td>
</tr>
<tr>
<td>10^{-4}</td>
<td>2</td>
<td>2  17  18  4  41</td>
<td>2.553  58.2</td>
<td>4.944  71.1</td>
</tr>
<tr>
<td>10^{-3}</td>
<td>17</td>
<td>3  92  28  38  95</td>
<td>0.015  58.2</td>
<td>0.031  71.1</td>
</tr>
<tr>
<td>10^{-2}</td>
<td>2935</td>
<td>201  141  59  112  512</td>
<td>0.000  58.2</td>
<td>0.001  71.1</td>
</tr>
</tbody>
</table>
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