Scaling Hardware Accelerated Network Monitoring to Concurrent and Dynamic Queries with *Flow

John Sonchack, Oliver Michel, Adam J. Aviv, Eric Keller, Jonathan M. Smith
Measuring High Speed Networks

100 Gb/s Links
>1 Tb/s Switches
Measuring High Speed Networks

Who is causing microbursts?
- Queue Lengths
- Drop Counts

100 Gb/s Links
>1 Tb/s Switches
Measuring High Speed Networks

Who is causing microbursts?
- Queue Lengths
- Drop Counts

Which flows are colliding?
- Utilization

Debugging

Traffic Engineering

100 Gb/s Links
>1 Tb/s Switches
Measuring High Speed Networks

Who is causing microbursts?
- Queue Lengths
- Drop Counts

Which flows are colliding?
- Utilization

Are any hosts compromised?
- Packet timing statistics

100 Gb/s Links
>1 Tb/s Switches
Measurement Challenges

Concurrent Applications    High Packet Rates

Who is causing microbursts?
- *Queue Lengths*
- *Drop Counts*

Which flows are colliding?
- *Utilization*

Are any hosts compromised?
- *Packet timing statistics*

- 3.2 Tbps
- > 100 M packets/s
- > 10 M flows/s
Prior Measurement Systems

Custom Hardware
(e.g., NetFlow)

Processing

Cluster
Prior Measurement Systems

Custom Hardware (e.g., NetFlow)

Flexibility
Applications can’t define custom statistics

Processing Cluster

Efficiency

Measurement Tput: < 1 M packets / second per core

Network Tput: > 100 M packets / second per switch
Prior Measurement Systems

Custom Hardware (e.g., NetFlow)
Flexibility

Processing Cluster
Efficiency

Reconfigurable Switch ASICS

[Marple SIGCOMM 17, TurboFlow EuroSys 18, Sonata SIGCOMM 18]

SELECT `packet.length` GROUP BY `tcp flow` AGGREGATE `average`
Prior Measurement Systems

Custom Hardware (e.g., NetFlow)
Flexibility

Processing Cluster
Efficiency

Reconfigurable Switch ASICS
Concurrency

[Marple SIGCOMM 17, TurboFlow EuroSys 18, Sonata SIGCOMM 18]

SELECT `packet.length`
GROUP BY `tcp flow`
AGGREGATE `average`

SELECT `packet.length`
GROUP BY `tcp flow`
AGGREGATE `average`
Concurrency Challenges with Reconfigurable ASICs

- Max Queue Length
- Total Drop Count
Concurrency Challenges with Reconfigurable ASICs

- Max Queue Length
- Total Drop Count
- Std Dev. Packet Size
- Avg. Flow Duration
- Avg. Utilization
- Jitter

Avg. Utilization

Avg. Jitter

Max Queue Len

Avg. Dur.

Drop Ct.

Firewall

L2 Forwarding

Std Dev Pkt Size
Concurrency Challenges with Reconfigurable ASICs

- Max Queue Length
- Total Drop Count
- Std Dev. Packet Size
- Avg. Flow Duration
- Avg. Utilization
- Jitter

Not Enough room in the datapath for all the queries.
Concurrency Challenges with Reconfigurable ASICs

Avg. Jitter

Firewall
L2 Forwarding
Max Queue Len
Avg. Dur.
Drop Ct.
Concurrency Challenges with Reconfigurable ASICs

Recompiling the datapath disrupts measurement and forwarding.

Recompile P4 ASIC at 29 seconds
*Flow: Efficiency and Concurrency?*

**Question:** can we leverage reconfigurable ASICs for concurrent monitoring?
*Flow: Efficiency and Concurrency?*

**Question:** can we leverage reconfigurable ASICs for concurrent monitoring?

**Insight:** concurrency challenges are caused by trying to do too much in the ASIC.
The Main Idea
The Main Idea

Preprocessing - Average Utilization Calculation

Preprocessing - Jitter Calculation

Preprocessing - Max Packet Length Calculation

SELECT packet.length
GROUP BY tcp flow
AGGREGATE max
The Main Idea

- Average Utilization Calculation
- Jitter Calculation
- Max Packet Length Calculation
The Main Idea

- Average Utilization Calculation
- Jitter Calculation
- Max Packet Length Calculation

Common Preprocessing
The Main Idea

Concurrency

Average Utilization Calculation
Max Packet Length Calculation
Jitter Calculation

Common Preprocessing
The Main Idea

- **Concurrency**
  - Average Utilization Calculation
  - Max Packet Length Calculation
  - Jitter Calculation
  - ...
The Main Idea

Concurrency

Efficiency

Average Utilization Calculation

Max Packet Length Calculation

Jitter Calculation

Common Preprocessing

…
The Main Idea

Intel’s new chip puts a teraflop in your desktop. Here's what that means.

It's as fast as a turn-of-the-century supercomputer.

By Rob Verger  June 1, 2017

Average Utilization Calculation
Max Packet Length Calculation
Concurrency Efficiency
The Main Idea

Concurrency

Efficiency

Measure every packet in > 1 Tb/s traffic with 1 server.

Average Utilization Calculation

Max Packet Length Calculation

Jitter Calculation

...
The Main Idea

- Application specific calculation
- Concurrency
- Preprocessing
- Efficiency

*Flow Cache
*Flow Agent
Outline

- Motivation
- Design
- Implementation
- Evaluation
*Flow Design*

```
SELECT ip length
GROUP BY tcp flow
AGGREGATE sum
```

1. Decoupling calculation
2. Generalize selection & grouping
Decoupling Calculation

SELECT *\text{ip length}\* GROUP BY *\text{tcp flow}\* AGGREGATE \text{sum}\*

<table>
<thead>
<tr>
<th>Flow Key</th>
<th>Length</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>A -&gt; B</td>
<td>200</td>
<td>1000</td>
</tr>
<tr>
<td>C -&gt; D</td>
<td></td>
<td>250</td>
</tr>
<tr>
<td>E -&gt; F</td>
<td></td>
<td>9416</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

Key: A -> B
Length: 200

Preprocessing

App. Specific Calculation

30
Decoupling Calculation

SELECT * 
GROUP BY * 
AGGREGATE * 

Grouped Packet Vector (GPV):
{flowKey: packetLength}

Preprocessing

App. Specific Calculation

Key: A -> B
Length: 200

<table>
<thead>
<tr>
<th>Flow Key</th>
<th>Pkt Len</th>
</tr>
</thead>
<tbody>
<tr>
<td>A -&gt; B</td>
<td>500 500 200</td>
</tr>
<tr>
<td>C -&gt; D</td>
<td>250</td>
</tr>
<tr>
<td>E -&gt; F</td>
<td>20 100 ...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Length Sum

1200
250
9416
...
Generalizing Selection and Grouping

```
SELECT ip length
GROUP BY host
AGGREGATE average
```

```
SELECT timestamp
GROUP BY tcp flow
AGGREGATE variance
```

Common Preprocessing → Grouped Packet Vectors (GPVs)
Generalizing Selection and Grouping

**SELECT ip length**
GROUP BY host
AGGREGATE average

**SELECT timestamp**
GROUP BY tcp flow
AGGREGATE variance

< flowKey:(packetLengths, packetTimestamps,...) >

Grouped Packet Vectors (GPVs)
Generalizing Selection and Grouping

SELECT ip, length
GROUP BY host
AGGREGATE average

SELECT timestamp
GROUP BY tcp flow
AGGREGATE variance

Regroup to any sub-key for 1 KV op. and copy per GPV.

export key = <src IP, dst IP, src port, dst port, protocol, link ID>

Common Preprocessing

Grouped Packet Vectors (GPVs)
*Flow Architecture

Packet feature selection and initial grouping.

Regrouping and statistics calculation.

Thin interface to grouped packet vectors.

*Flow Cache

*Flow Agent

Monitoring Applications

Grouped Packet Vectors
Outline

• Introduction
• Design
• Implementation
• Evaluation
*Flow Implementation

Proof-of-concept implementation.

P4 implementation for 3.2 Tb/s Barefoot Tofino.

*Flow Cache

Grouped Packet Vectors

*Flow Agent

Monitoring Applications
Background:
Stateful Match Action ASICS

Restrictive computational model

Line rate processing (1 packet per cycle)

Packet Headers + metadata

Match | Action | Stateful Variables
**Flow as a Stateful Match Action**

Pipeline

<table>
<thead>
<tr>
<th>Slot ID</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
</tr>
<tr>
<td>3</td>
<td>c</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Slot ID</th>
<th>Packet Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Slot ID</th>
<th>Packet Features Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20 1400</td>
</tr>
<tr>
<td>2</td>
<td>77</td>
</tr>
<tr>
<td>3</td>
<td>19 500 135 32</td>
</tr>
</tbody>
</table>

*Key: d*  
*Size: 14*  
*Packet Headers + metadata*
*Flow as a Stateful Match Action Pipeline

<table>
<thead>
<tr>
<th>Slot ID</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
</tr>
<tr>
<td>3</td>
<td>c</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Slot ID</th>
<th>Packet Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Slot ID</th>
<th>Packet Features Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20 1400</td>
</tr>
<tr>
<td>2</td>
<td>77</td>
</tr>
<tr>
<td>3</td>
<td>19 500 135 32</td>
</tr>
</tbody>
</table>
*Flow as a Stateful Match Action Pipeline

**Untracked Flow:**
Replace and copy to software

---

**Key:** d
**Size:** 14

Packet Headers + metadata

---

<table>
<thead>
<tr>
<th>Slot ID</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
</tr>
<tr>
<td>3</td>
<td>c</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Slot ID</th>
<th>Packet Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Slot ID</th>
<th>Packet Features Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20 1400</td>
</tr>
<tr>
<td>2</td>
<td>77</td>
</tr>
<tr>
<td>3</td>
<td>19 500 135 32</td>
</tr>
</tbody>
</table>
*Flow as a Stateful Match Action Pipeline*

Untracked Flow: Replace and copy to software

Key: d
Size: 14
Packet Headers + metadata

Key: a
Tuples: [20, 1400]

Slot ID | Key  
-------|------
1      | d    
2      | b    
3      | c    

Slot ID | Packet Count 
-------|-------------
1      | 1           
2      | 1           
3      | 4           

Slot ID | Packet Features Buffer 
-------|-----------------------
1      | 14        
2      | 77        
3      | 19 500 135 32
*Flow as a Stateful Match Action Pipeline*

**Untracked Flow:**
Replace and copy to software

**Tracked Flow:**
Append tuple

**Key:** $b$
**Size:** 8

**Packet Headers + metadata**

**Table:**

<table>
<thead>
<tr>
<th>Slot ID</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>d</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
</tr>
<tr>
<td>3</td>
<td>c</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Slot ID</th>
<th>Packet Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Slot ID</th>
<th>Packet Features Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>77 8</td>
</tr>
<tr>
<td>3</td>
<td>19 500 135 32</td>
</tr>
</tbody>
</table>
*Flow as a Stateful Match Action Pipeline*

**Untracked Flow:** Replace and copy to software

**Tracked Flow:** Append tuple

**Tracked Flow, Buffer Full:** rollover buffer

---

**Key:** c  
**Size:** 10  
**Packet Headers + metadata**

---

<table>
<thead>
<tr>
<th>Slot ID</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>d</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
</tr>
<tr>
<td>3</td>
<td>c</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Slot ID</th>
<th>Packet Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Slot ID</th>
<th>Packet Features Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>77 8</td>
</tr>
<tr>
<td>3</td>
<td>19 500 135 32</td>
</tr>
</tbody>
</table>
*Flow as a Stateful Match Action Pipeline*

<table>
<thead>
<tr>
<th>Slot ID</th>
<th>Key</th>
<th>Packet Count</th>
<th>Packet Features Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>d</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>b</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>c</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Untracked Flow: Replace and copy to software

Tracked Flow: Append tuple

Tracked Flow, Buffer Full: rollover buffer

Key: c
Size: 10

Packet Headers + metadata

Key: a
Tuples: [19, 500, ...]
*Flow as a Stateful Match Action

Pipeline

Untracked Flow: Replace and copy to software

Tracked Flow: Append features

Tracked Flow, Buffer Full: rollover buffer

Untracked Flow:

Tracked Flow:

Key: a
Tuples: [20, 1400]

Key: c
Tuples: [19, 500, ...]

Packet Headers + metadata

Flow: 1
Key: a
Tuples: [20, 1400]

Flow: 2
Key: b

Flow: 3
Key: c
Tuples: [19, 500, ...]

Packet Count

Packet Features Buffer

1  2  1
2  2
3  1

1  14 1400
2  77  8
3  49 500 300 143
Outline

• Motivation
• Design
• Implementation
• Evaluation
Evaluation: Concurrency and Efficiency
Evaluation: Concurrency and Efficiency

*Flow Cache

Stages  ALUs

Query  Stages  ALUs
Connection Count
Avg. Latency
Flowlet Sizes
Packet Counts
TCP Non-Monotonic
TCP Out of Sequence

*Flow Cache

GPV

Vs

Compiled Queries

[Marple SIGCOMM 17]
Evaluation: Concurrency and Efficiency

The *Flow Cache can service any query but only requires as many hardware resources as \( \sim 1 \) compiled query.

*Flow Cache

<table>
<thead>
<tr>
<th>*Flow Cache</th>
<th>Stages</th>
<th>ALUs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11</td>
<td>33</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query</th>
<th>Stages</th>
<th>ALUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Count</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Avg. Latency</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>Flowlet Sizes</td>
<td>11</td>
<td>31</td>
</tr>
<tr>
<td>Packet Counts</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>TCP Non-Monotonic</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>TCP Out of Sequence</td>
<td>7</td>
<td>14</td>
</tr>
</tbody>
</table>

Compiled Queries

[Marple SIGCOMM 17]
Evaluation: Concurrency and Efficiency

- Host Profiler: Classifies TCP flows based on 12 features.
- Flow Classifier: Generates per-host timestamp vectors.
- Congestion Debugger: Identifies end hosts with packets in queue during microburst.

Throughput (M GPVs/s)

100 X average GPV rate of 10 GbE Internet link [3].

[3]: CAIDA 2015 and 2016 trace stats
https://www.caida.org/data/passive/trace_stats/
Evaluation: Concurrency and Efficiency

One GPV processing server can measure terabit rate traffic.

- Host Profiler: Classifies TCP flows based on 12 features.
- Flow Classifier: Generates per-host timing profiles.
- Congestion Debugger: Identifies end hosts with packets in queue during microburst.

Throughput (M GPVs/s)

100 X average GPV rate of 10 GbE Internet link [3].

[3]: CAIDA 2015 and 2016 trace stats
https://www.caida.org/data/passive/trace_stats/
Outline

• Motivation
• Design
• Implementation
• Evaluation
In The Paper

On Github

www.github.com/jsonch/starflow

ASIC Resource Requirements and More Evaluation

Table 3: Average throughput, in GPPs per second, for *Flow agent and applications.

<table>
<thead>
<tr>
<th></th>
<th>Efficient</th>
<th>Flexible</th>
<th>Concurrent</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Netflow</td>
<td>✔</td>
<td>x</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Software</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>PFE Queries</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>x</td>
</tr>
<tr>
<td>*Flow</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

Match+Action Layout

3.2 Tb/s *Flow Cache Prototype for Barefoot Tofino

```
// Tables
table updateKey { default_action :UpdateKeyAction(); }
table updateFeatures { default_action :UpdateFeaturesAction(); }
table resetFeatures { default_action :ResetFeaturesAction(); }

// Actions.
// Update key for every packet.
```
Conclusion
(and Thank You for Listening!)

What are the right measurement subtasks to do in hardware?

- Concurrency
- Efficiency

www.github.com/jsonch/starflow

*Flow Applications*

- Statistics Calculation

*Flow Cache*

- Selection and Grouping