NetBOA: Self-Driving Network Benchmarking

Andreas Blenk*

Joint work with:
Johannes Zerwas*, Patrick Kalmbach*, Laurenz Henkel*, Gábor Rétvári^, Wolfgang Kellerer*, Stefan Schmid°

*Technical University of Munich, Germany
^Budapest University of Technology and Economics, Hungary
°Faculty of Computer Science, University of Vienna, Austria

Workshop on Network Meets AI & ML (NetAI’19)
Beijing, China, 2019
Today’s Approach of Operating Networks?

With more complex networks need for automation!
What Self-Driving Networks Should Do

- Self-Monitoring
- Self-Optimizing
- Self-Benchmarking

Network Problem

Solution

Empowerement (SelfDN’18)

Performance Evaluation

NetBOA (NetAI’19)

O’zapft is (BIG DAMA’17)

Source: https://www.pinterest.at/pin/318137161149129652/
Benchmarking Network Algorithms, Architectures etc…
The Traditional Way …

Not always available
Traces

Not generalizing
Models

Hmm… Biased?
Human’s Best Guesses

Data-Driven

This Talk: Use Machine Learning to Benchmark Networks (or more concrete network functions) …
What Could be Seen as Related

- Algorithmic complexity attacks (software domain):
  - SlowFuzz
  - PerfFuzz
- Automated Synthesis of Adversarial Workloads for Network Functions, ACM Sigcomm 2018
- Policy Injection: A Cloud Dataplane DoS Attack, ACM Sigcomm DEMO 2018

Why Important?

Implementation aspects can harm performance

Could even be used to attack your systems!

We propose NetBOA to automatically create network traffic input
Use Case: Benchmark Open vSwitch

Objective: Find Network Traffic Configuration that Maximizes CPU/Latency
Network Benchmarking is Challenging: Complex and Huge Configuration Space

How many packets to send? How should headers look like? What protocol to use? When to send packets? Etc.

- Number of Network Packets [1000 – 5000]
- Batch Size [1-5]
- Packet Inter Arrival Time [1ms – 13ms]
- VLANs [1-5]

Human still Involved!
NetBOA: The Bayesian Optimization Measurement Loop

1. Set configuration
2. Measure until confidence is reached
3. New measurement points
4. Machine Learn Performance Model

NetBOA
Bayesian Optimization
Acquisition Function
Maximize Expected Improvement
Update Posterior
Fit Gaussian Process

Number of packets
Inter arrival times [milliseconds]

CPU \[%\]
Bayesian Optimization: NetBOA for Inter Arrival Time (IAT) Parameter

Update Gaussian Process at runtime

Sampling from Gaussian Process gives confidence

Evaluation: Compare NetBOA with GridSearch and RandomSearch
Performance models are non-trivial

**Surprising:** Sending less network packets over time can lead to significantly higher CPU usage.
We are using the OvS switch with the **Megaflow Cache enabled**

- For instance for 5000 packets: We trigger roughly every >2 ms a flow insertion + removal
  → Forcing OvS to continuously run through the array + resizing it

**Why? Let Us Look At OvS Behavior!**

**OvS rule timeout 10 seconds**

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Drop</td>
</tr>
<tr>
<td>2</td>
<td>Drop</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Drop</td>
</tr>
<tr>
<td>2</td>
<td>Drop</td>
</tr>
<tr>
<td>3</td>
<td>Drop</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>N</td>
<td>Drop</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Drop</td>
</tr>
<tr>
<td>2</td>
<td>Drop</td>
</tr>
<tr>
<td>3</td>
<td>Drop</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>N</td>
<td>Drop</td>
</tr>
</tbody>
</table>

Every packet triggers 2 times a costly array resizing operation!
NetBOA vs Random Search

NetBOA

Better
Faster

Random Search

24 % higher CPU utilization
Conclusion

- Summary: NetBOA is a Bayesian Optimization-based data-driven approach to generate network traffic configurations for benchmarking network function implementations

  - NetBOA can efficiently find challenging network traffic configurations (maximize CPU/Latency)
  - NetBOA can also be used to minimize, e.g., CPU or Latency

- Open questions and problems:
  - Does beating the machine means it generalizes?
  - Does it scale?
  - Alternatives?
  - Bayesian Optimization needs also tuning!
References

[BIG DAMA’17] Blenk, Andreas; Kalmbach, Patrick; Schmid, Stefan; Kellerer, Wolfgang: o'zapft is: Tap Your Network Algorithm's Big Data! ACM SIGCOMM 2017 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks (Big-DAMA), 2017

[SelfDN’18] Kalmbach, Patrick; Zerwas, Johannes; Babarczi, Péter; Blenk, Andreas; Kellerer, Wolfgang; Schmid, Stefan: Empowering Self-Driving Networks. Proceedings of the Afternoon Workshop on Self-Driving Networks - SelfDN 2018, ACM Press, 2018

Thank you!

Questions?