Avenir: Managing Data Plane Diversity via Control Plane Synthesis

Eric Hayden Campbell
Bill T. Hallahan, Priya Srikumar, Carmelo Cascone, Jed Liu, Vignesh Ramamurthy, Hossein Hojjat, Ruzica Piskac, Robert Soulé, Nate Foster

NSDI 2021
Engineers work at Homogenous Abstraction Layer

Control Plane abstraction implemented by hardware vendors

Traditional Network
Engineers manage data plane diversity

Software Defined Network

<table>
<thead>
<tr>
<th>Ip.Dst</th>
<th>Ip.Src</th>
<th>port</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.1</td>
<td>10.0.1.0</td>
<td>101</td>
</tr>
</tbody>
</table>

Diverse Data Plane

<table>
<thead>
<tr>
<th>Ip.Dst</th>
<th>Meta</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.1</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ip.Src</th>
<th>Meta</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.1.0</td>
<td>3</td>
<td>101</td>
</tr>
</tbody>
</table>
A New Driver for Every Target

Abstract Operations

Driver

Target Operations

Target Switch

Abstract Switch

<table>
<thead>
<tr>
<th>src</th>
<th>port</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.1</td>
<td>88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>dst</th>
<th>port</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.1</td>
<td>101</td>
</tr>
</tbody>
</table>

void add(src, port) {
    m = PortMap.lookup(port);
    l2.add(*, src, m);
    agg.add(m, port)
}

12.add(10.0.1.0, 88);
12.add(*, 10.0.0.1, 47);
agg.add(4, 88)

<table>
<thead>
<tr>
<th>ip.dst</th>
<th>ip.src</th>
<th>m</th>
<th>port</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.1</td>
<td>*</td>
<td>3</td>
<td>101</td>
</tr>
<tr>
<td>*</td>
<td>10.0.1.0</td>
<td>47</td>
<td>88</td>
</tr>
</tbody>
</table>
Managing Drivers is Tedious

<table>
<thead>
<tr>
<th>ip.src</th>
<th>port</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.1</td>
<td>CTRL</td>
</tr>
<tr>
<td>10.0.0.1</td>
<td>101</td>
</tr>
</tbody>
</table>

Challenges

- State Management
- Scale
- Complexity
- Programmable Pipelines
- (Under)-specification

Qumran-MX 2 years to production
Goal:
Automatic Translation of Control Plane Operations

Subgoal: Verified

Subgoal: Efficient
Avenir: Control Plane Synthesis

Abstract Operations → Avenir → Target Operations

\( \varphi_{\text{abs}} \) → Avenir → \( \varphi_{\text{tgt}} \)

Fully Automatic!

Formally Verified!

Abstract Operations:

```
... vld.apply(); 12.apply(); 13.apply(); ...
```

Target Operations:

```
... 12.apply(); vld2.apply(); 13.apply(); vld3.apply(); ...
```
Diverse Data Plane

Homogenous Abstract Data Plane

Avenir

Diverse Data Plane
Controller

Homogenous Abstract Data Plane

Diverse Data Plane
Synthesis

Counter-Example Guided Inductive Synthesis (CEGIS)

Theorem (**Soundness**). Synthesized operations correctly realize abstract behavior.

Proof. Following Dijkstra ‘75. QED.

Theorem (**Completeness**). If a solution exists, Avenir **eventually** computes it.

Proof. By the finiteness of (intractably large) search space. QED.
Making CEGIS Fast via Incremental Synthesis

- Controllers make small frequent changes
  - Process a single abstract operation at a time
  - Optimize Avenir for this common case

- Assume abstract and target behaviors are equal
  - Program Slicing
  - Static Analysis
  - Configurable Domain Specific Heuristics

Verification

Heuristic Search

Avenir

Abstract Operation

src.add(10.0.1.0, DROP)

Agg.add(4, 88)

Counter-example
(ip.src = 10.0.1.0)

Target Operations

Target

12.add(*, 10.0.1.0, 4);
agg.add(4, 88)

Avenir

Target Operations

Target

12.add(*, 10.0.1.0, 4);
agg.add(4, 88)
Making CEGIS Fast via Abstracting Caches

- **Template Cache**
  - Infers structure from previous translations
  - Replicates mapping of keys & action data
  - `src.add(ipv4.src, port)`
    \[ \rightarrow 12.\text{add}(\ast, \text{ipv4.src}, m); \text{agg.add}(m, \text{port}) \]

- **Query Cache**
  - Generalize concrete values in queries
  - \( x = 5 \lor x \neq 5 \) becomes \( \forall b. x = b \lor x \neq b \)
  - Checking \( x = 47 \lor x \neq 47 \) is purely syntactic
How Broad is Avenir?

• Retarget One Abstract L2/L3/Validate Pipeline
• … to various handwritten target pipelines:

Takeaways
• Support diverse target pipelines
• Caches amortize the cost of learning
• Pre-populate caches for lightning-fast speeds
How Well Does Avenir Scale?

Full Program Scaling

Classifier Scaling

Takeaways
- Seems to scale exponentially
- Efficient on common program sizes
- *Results dependent on heuristics
How Efficient is Avenir In A Realistic Scenario?

- End-to-end ONOS switch reboot load test
- 40k IPv6 Routes

40k Routes Completion

Takeaway: Same Order of Magnitude
Thank you

This work is supported by NSF GRFP, NSF CCF, NSF FMiTF, and DARPA grants as well as gifts from Alibaba, Fujitsu, Infosys, and VMware.