The Beacon OpenFlow Controller
www.beaconcontroller.net
David Erickson
Stanford
Motivation

• Back to circa 2008-2009
• The OpenFlow controller world == NOX
  – Single threaded event based C++ with SWIG glue to Python
  – Python apps
  – C++ apps
Great! ...

- ... but room for improvement

- Python
  - Inconsistent API between C++/Python (SWIG)
  - Significantly slower than C++

- C++
  - Slow compilation
  - Cryptic compilation errors (STL, templates)
  - Manual memory management
  - Linux only, significant list of dependencies
Questions

• Can I contribute solutions to these issues to NOX?
  – Yes to some.
  – No to others due to programming language.

• Am I going to be using a controller platform for awhile?
  – Yes.

• Should I try and build one that improves on these problems?
  – Probably.
  – *Sigh*. 
One more thing...

• Other useful features
  – Runtime Modularity
  – Fast, Fully Multithreaded
Improvement Summary

• Language Specific
  – Slow compilation
  – Cryptic compilation errors (STL, templates)
  – Manual memory management
  – Linux only, significant list of dependencies

• Implementation
  – Runtime Modularity
  – Fast, Fully Multithreaded
# Language Exploration

<table>
<thead>
<tr>
<th>Language</th>
<th>Fast Compilation</th>
<th>Managed Memory</th>
<th>Cross Platform</th>
<th>High Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>C#</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>Java</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>?</td>
</tr>
<tr>
<td>Python</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
</tbody>
</table>
Why Modularity?

• Code level
  – Interfaces, Implementations

• Start Time
  – Select apps to run

• Run Time
What is Runtime Modularity?

• SDN Controller \(\sim=\) Operating System
  – Stop, Start, Install, Remove Apps at Runtime

• Uses
  – Application restart
  – Online App Store
  – Live Updates
  – Debugging

• Enabled by OSGi
Beacon

Northbound

Applications
- Learning Switch
- Topology
- Device Mgr
- Routing
- Web UI
- Your App!

Core

Southbound

OpenFlow

Your App!
Performance

- Each app gets OFMMessages from all threads
Read Designs

• Run to Completion

I/O Threads

App 1
App 2
App 3

• Shared Queue

I/O Threads

Pipeline Threads

App 1
App 2
App 3

Shared Queue
Write Designs

- **Immediate**
  - Application:
  - Core (I/O Loop):
  - Kernel:

- **Batched**
  - Application:
  - Core (I/O Loop):
  - Kernel:

❗️ - Write

Time
Performance

• Cbench
  – Run on EC2, cluster compute instance
  – Easily reproducible
  – Throughput mode

<table>
<thead>
<tr>
<th>Write Path</th>
<th>Read Path</th>
<th>Shared Queue</th>
<th>Run to Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Immediate</td>
<td></td>
<td></td>
<td>Beacon Imm.</td>
</tr>
<tr>
<td>Batched</td>
<td>Beacon Queue</td>
<td>Beacon</td>
<td>Beacon</td>
</tr>
</tbody>
</table>
Single Threaded Controllers

![Bar chart showing performance comparisons between different controllers]

<table>
<thead>
<tr>
<th></th>
<th>Shared Queue</th>
<th>Run to Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beacon</td>
<td>Beacon Queue</td>
<td>Beacon Imm.</td>
</tr>
<tr>
<td>Beacon Imm.</td>
<td>Maestro</td>
<td>NOX</td>
</tr>
<tr>
<td>Floodlight</td>
<td></td>
<td>POX</td>
</tr>
<tr>
<td>Maestro</td>
<td></td>
<td>Ryu</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Write Path</th>
<th></th>
<th>Read Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Immediate</td>
<td></td>
<td>Beacon Imm.</td>
</tr>
<tr>
<td>Batched</td>
<td>Beacon Queue</td>
<td>Beacon</td>
</tr>
</tbody>
</table>
Multithreaded Controllers

![Graph showing performance metrics for multithreaded controllers. The x-axis represents the number of threads, ranging from 2 to 12. The y-axis represents responses per second, ranging from 100K to 10M. The graph includes lines for Beacon, Beacon Queue, Beacon Imm., Floodlight, Maestro, NOX, and Immediate. The table below the graph lists the shared queue and run to completion for immediate and batched operations.](image)
Conclusions

• Productivity++ ✓
• Runtime Modularity ✓ ✓
• Performance ✓
• Open Source Progeny
  – 2012 – Floodlight
    • http://www.projectfloodlight.org/floodlight/
  – 2013 – OpenDaylight Controller
    • http://www.opendaylight.org/
Questions?

www.beaconcontroller.net

David Erickson

Stanford