Erlang, Open Networking, and the Future of Computing

Stu Bailey, Founder/CTO
What is the Business View of the Network?

Traditional corporate network

Business accountable network
How Can We Program an Internet of Things?
How Can We Program a Million Cores?
And We Must Assume Hardware is Failing (or changing) ALL the Time: “Write Once, Run Forever”
OF COURSE ERLANG
But Wait! What About the Network?!
Of Course
Infiniband!
Except...
And the Winner Is...
A STANDARD Forwarding Plane and Logically Centralized Control Plane

Capabilities across forwarding plane vendors are fairly uniform. Performance and capacity are primary differentiators.
FlowForwarding.org

OPEN SOURCE SDN STACK
ENTERPRISE FOCUSED

...and maybe more?
Full SDN Stack Timeline

FlowForwarding Community
+ alpha LINC Switch code availability

LINC Switch beta v2.X

LINCX Switch v1.0

LINC Switch beta v1.0

Loom Control Plane + Tapestry PoC

Loom Scalable SDN Control Plane
Erlang/OTP = SDN + Big Data + IoT
LINC Switch Architecture

OpenFlow Capable Switch

OpenFlow Operational Context

OpenFlow Logical Switch

Flow Table

Secure Channel

Group Table

Flow Table

Pipeline

OpenFlow Resource

OpenFlow Protocol

OF-Config

OpenFlow Controllers

OpenFlow Controllers

OpenFlow Capable Switch

OpenFlow Protocol

OpenFlow Configuration Points
Erlang Implementation Architecture

[Diagram showing the architecture with Erlang VM, OF-Config, OpenFlow Protocol, and corresponding modules: of_configl, of_cfg_channel, of_cswitch, of_protocol, of_channel, of_switch, and API.]
LINCX

Maxim Kharchenko, Cloudozer LLP
LINCX Architecture

- Dom0 (Linux)
- Xen Hypervisor
- Physical Host: Eth0, Eth1, Eth2, Eth3, Eth4, EthN
The Erlang Secret Sauce

Controller

flowmods via OpenFlow (1.3)

Conversion to Erlang AST

Erlang compiler

Match engine

Dom0 (Linux)

Xen Hypervisor

Physical Host

Eth0 Eth1 Eth2 Eth3 Eth4 EthN

LINCX

LING

Infoblox

© 2014 Infoblox Inc. All Rights Reserved.
Future

Multicore

• A switch per port? Per two ports?

Faster software bridges

• Replace generic Linux bridges with faster alternative

PCI passthrough

• Eliminate Linux completely from the packet path
The State of the Art…is Not

<table>
<thead>
<tr>
<th>Data Platforms</th>
<th>Distributed Data Intensive Applications (e.g. Hadoop, Apache Spark, Sensor Nets)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual System Abstractions</td>
<td>OS (e.g. Linux), Abstract Machines (e.g. JVM), Languages (e.g. Ruby), Clustering Frameworks (e.g. OpenStack), Databases/FileSystems (e.g. MongoDB, HDFS), HDN++</td>
</tr>
<tr>
<td>Fundamental Abstractions</td>
<td>Ethernet Frame</td>
</tr>
<tr>
<td>Increasingly Ubiquitous Hardware</td>
<td>Network Processors (Broadcom, Qualcomm)</td>
</tr>
</tbody>
</table>
The Right Perspective...and Perfect Timing

Ethernet + Multi-Core = A New Ubiquitous Machine
The Right Perspective...and Perfect Timing

"The network IS the computer"

John Gage (1984)
### We Need a Better Distributed System!

<table>
<thead>
<tr>
<th>Scalable System Abstractions</th>
<th>Specific Distributed Apps</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Data Intensive Distributed Control Plane) SDN + Distributed GraphDB + Functional Language</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fundamental Abstractions</th>
<th>Scalable Analytic Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet Frame</td>
<td>Virtual Machine (Xen)</td>
</tr>
</tbody>
</table>

| Increasingly Ubiquitous Hardware | |
|---------------------------------||
| Network Processors (Broadcom, Qualcomm) | Microprocessors (x86, ARM) |
Great But...

WhatsApp?
Is there a SINGLE NUMBER that captures NETWORK COMPLEXITY?

Easy to COMPUTE?
Easy to UNDERSTAND?
**DETECT COMMUNITIES**

**THE NETWORK COMPLEXITY INDEX**

A measure which rises with an increase in the number of LARGE DISTINCT ACTIVITIES on a given network.

*LARGE DISTINCT ACTIVITY:*

Lots of participants interacting toward a common goal.
Please connect me to GOOGLE.
My # is (a, b, c, d); Google’s # is (w, x, y, z).

DNS query: Where is GOOGLE?

“Google?”

“Recursiver
DNS

“I know!
Google’s # is... (w, x, y, z)”

TIME
(hh:mm)

CALLER #
(a, b, c, d)

CALLED #
(w, x, y, z)

ROUTER

“Yes,
this is Google.
Go ahead.”

ROUTERS
work out
the PATH
to and from
Google.
The NETWORK COMPLEXITY INDEX (NCI) is defined as:

\[ B(N) = \text{Max } j, \ X[j] \geq j \]

The BALANCE POINT between the SIZE of network activities and the NUMBER network activities.

“LARGEST” activity (the most participating endpoints)

“SMALLEST” activity (the FEWEST participating endpoints)

MANY \leftrightarrow \# \text{ of ENDPOINTS} \leftrightarrow FEW

DYNAMIC NETWORK WIDE PROPERTY
GLOBAL TELEMETRY AND ACTION
LOGICAL INSTEAD OF PHYSICAL
A NEW Machine?

IF

= 

THEN
We Can Look Past the Cloud...
How Can an Erlang/OTP Developer Program the Entire Network at Runtime?

What abstractions?

Mechanisms?

Language extensions?

ETC…
Is I/O really a side effect?
Is computation king?
Or is Communication Just as Fundamental?

Where are Joe’s Contract Checkers??
Some Grand Challenges to Modernize Distributed Erlang

- Security (e.g. Internode Security)
- Modernize Distribution with SDN
- Scalable GUI, Testing, and Diagnostic Frameworks
- Packaging and Code Maintenance
- Million-clause Functions, Modern Mailboxes, etc.

100% Erlang SDN at EUC 2015!
Help us make it happen!

Email us @ info@FlowForwarding.org
http://www.FlowForwarding.org