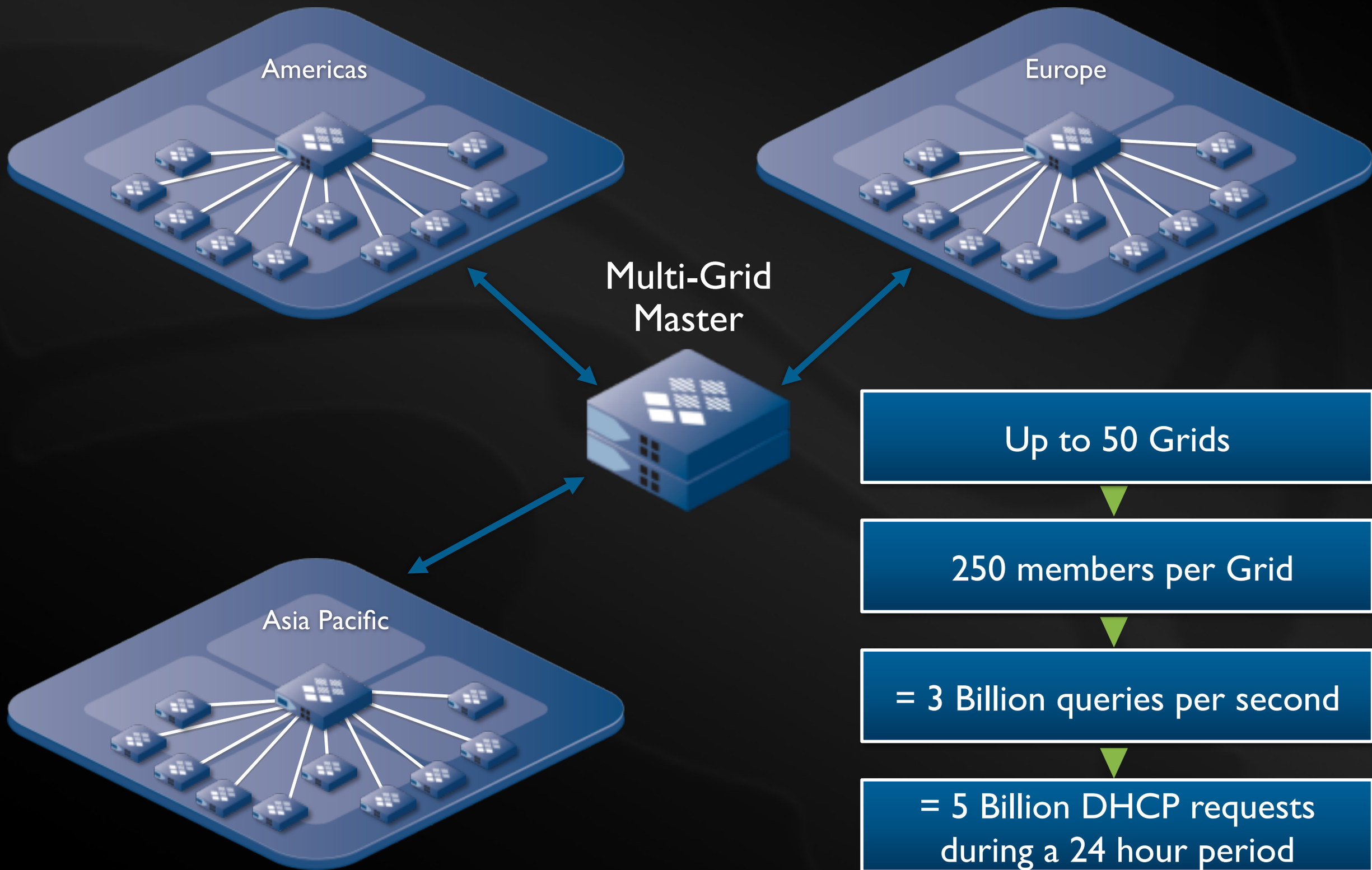
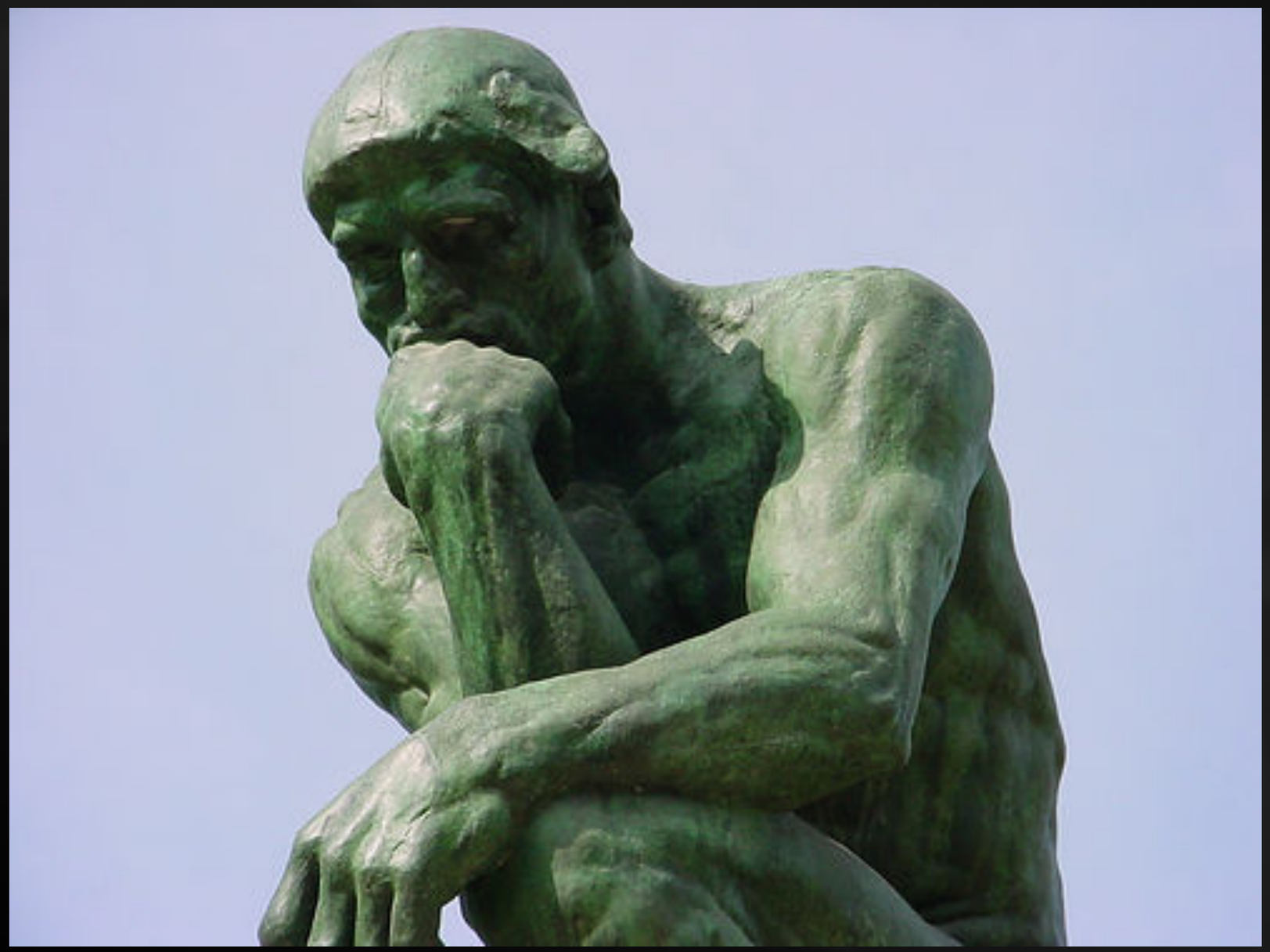
Erlang and OpenFlow: A Match Made in The Cloud!

Stuart Bailey
Founder/CTO Infoblox
Erlang Factory March 2013

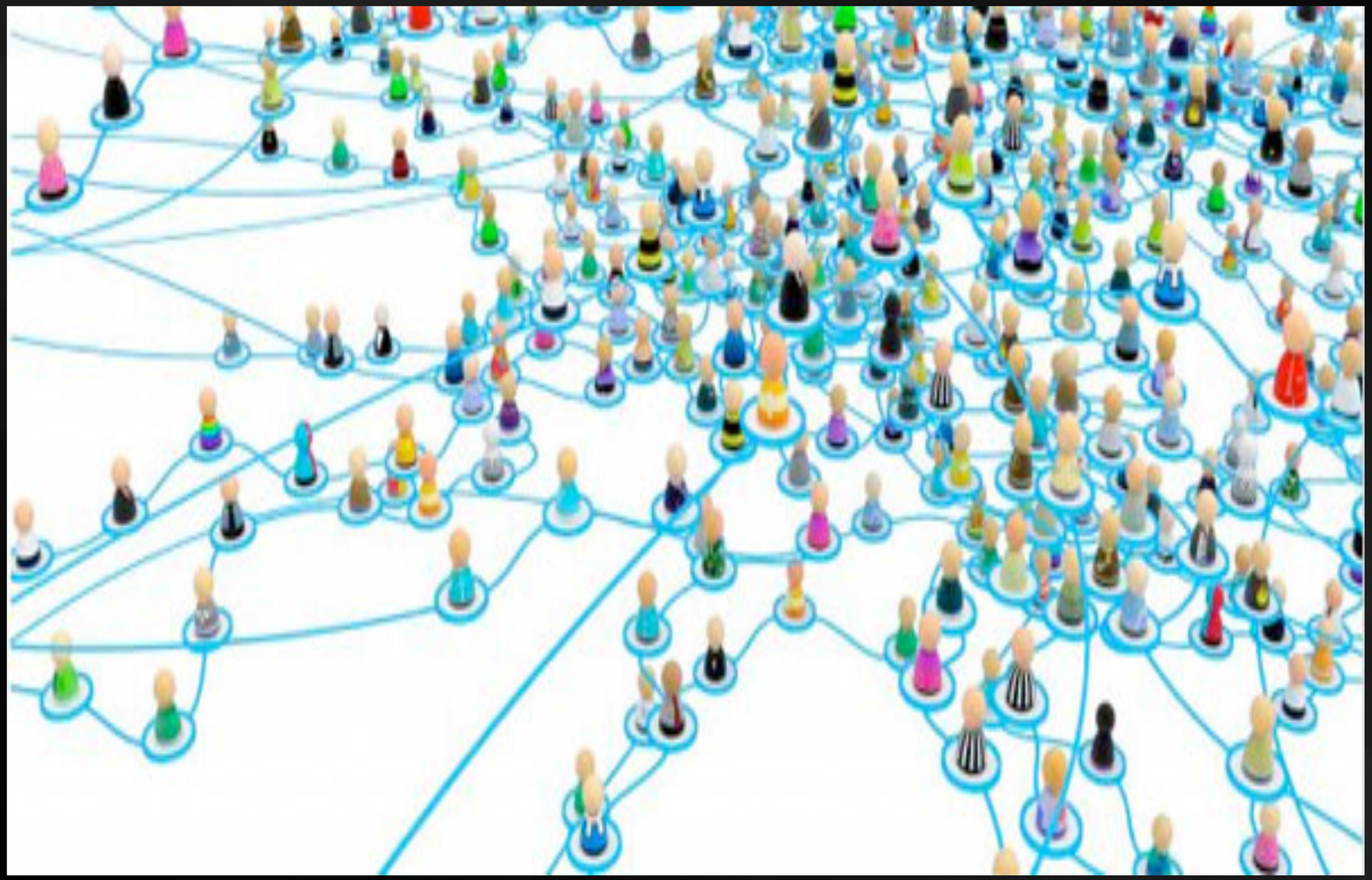
Imagine a large data PROCESSING platform (Hadoop?)



How do we scale this system?



To an Internet of Things

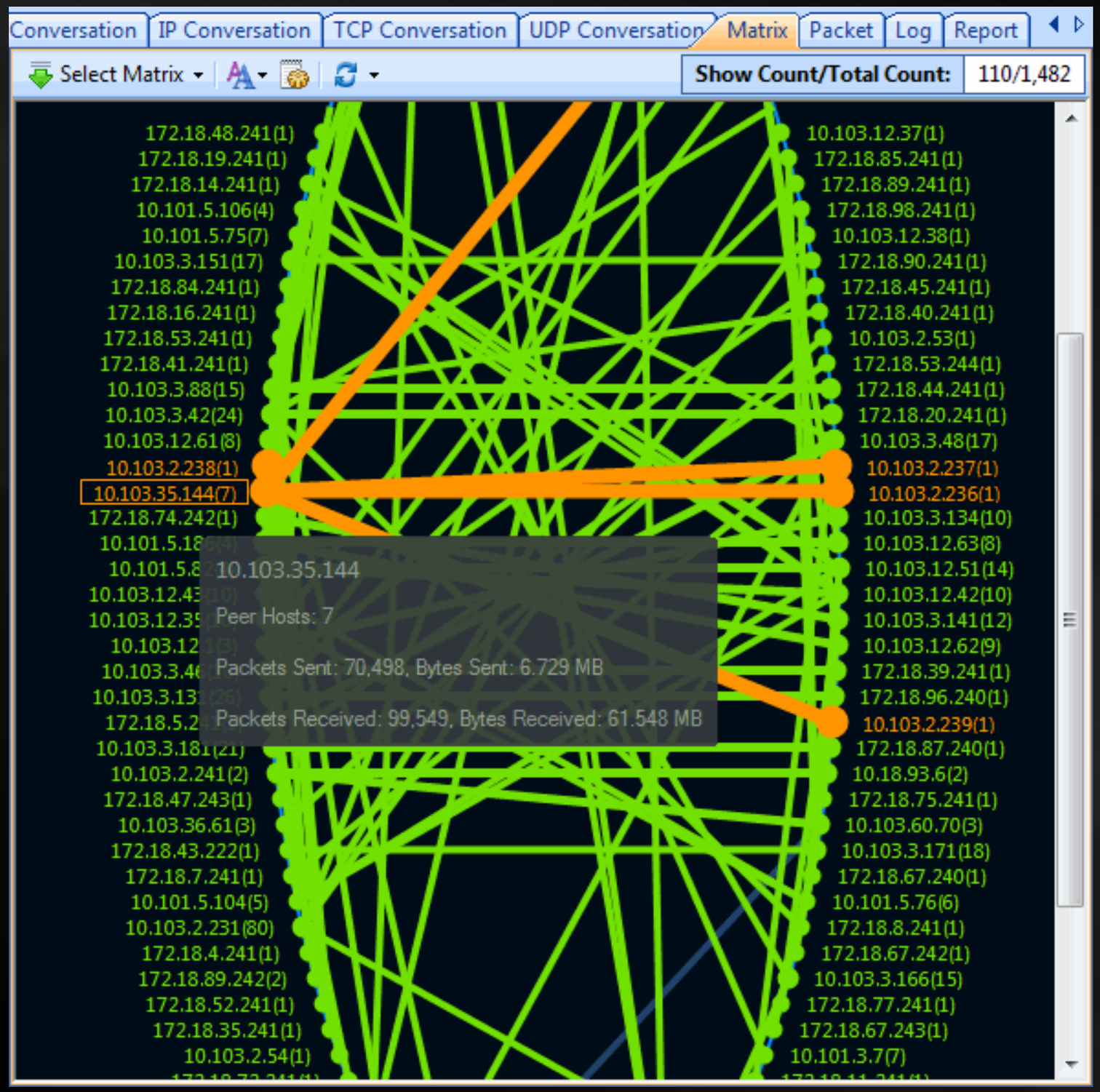


To a Million Cores!

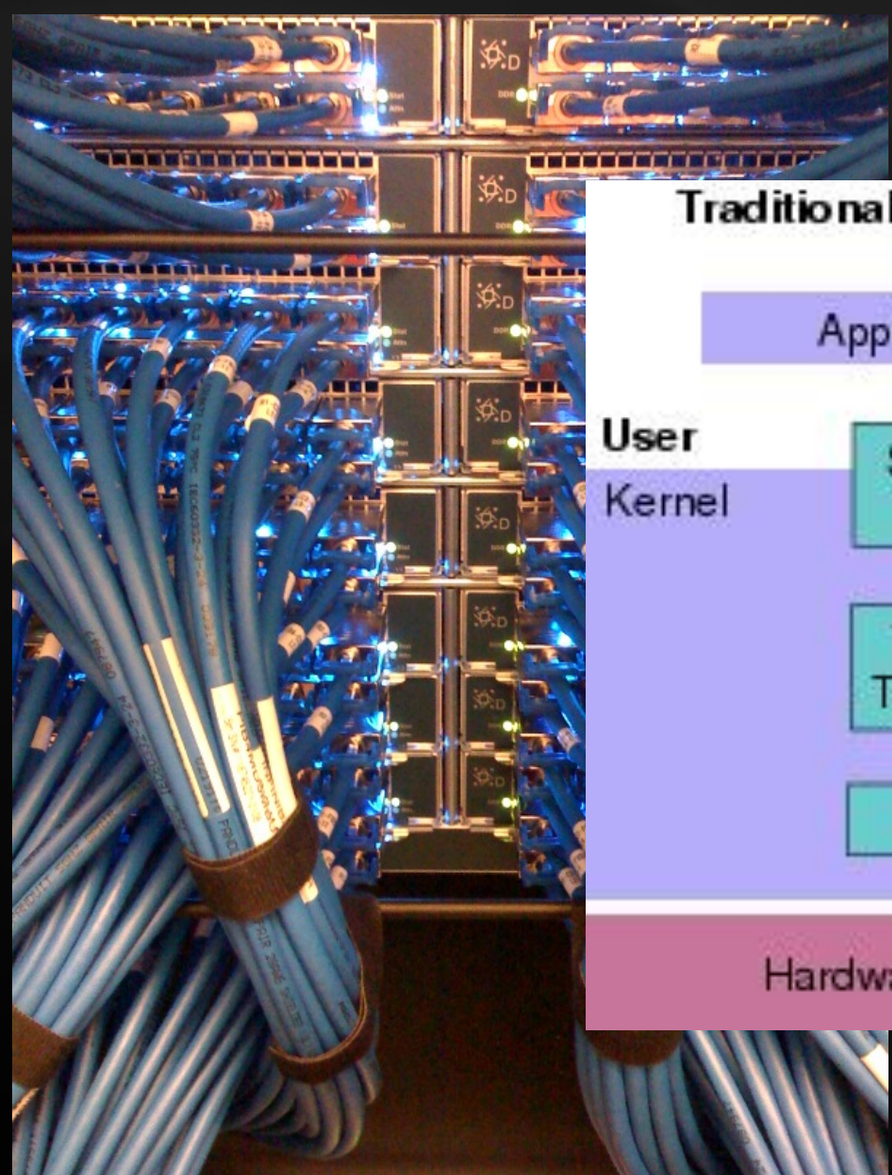
Assume hardware is failing (or changing) ALL the time: “write once run forever”

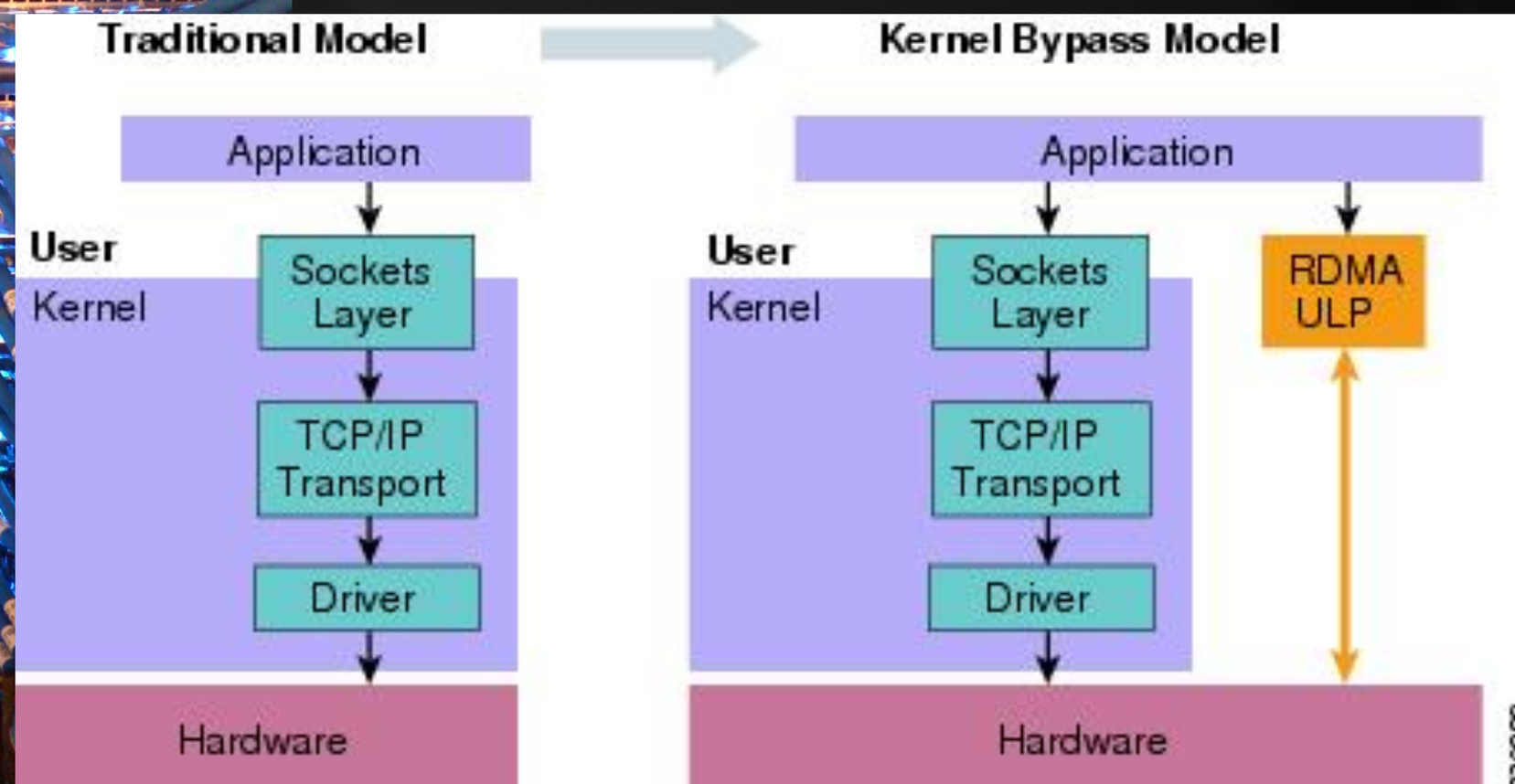
Of Course Erlang!!

But wait! What about the network?!



Of course Infiniband! Except…



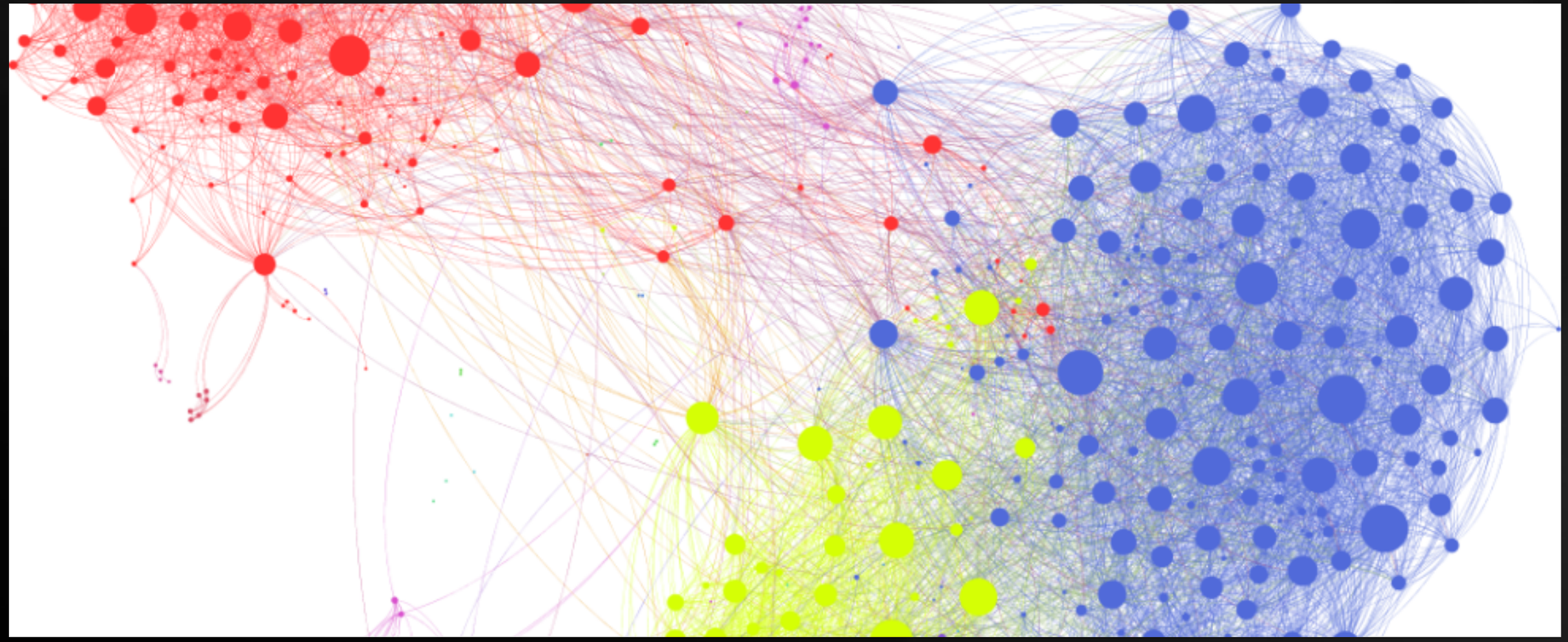


Infiniband is cost prohibitive because it's not likely to commoditize at the right scale…

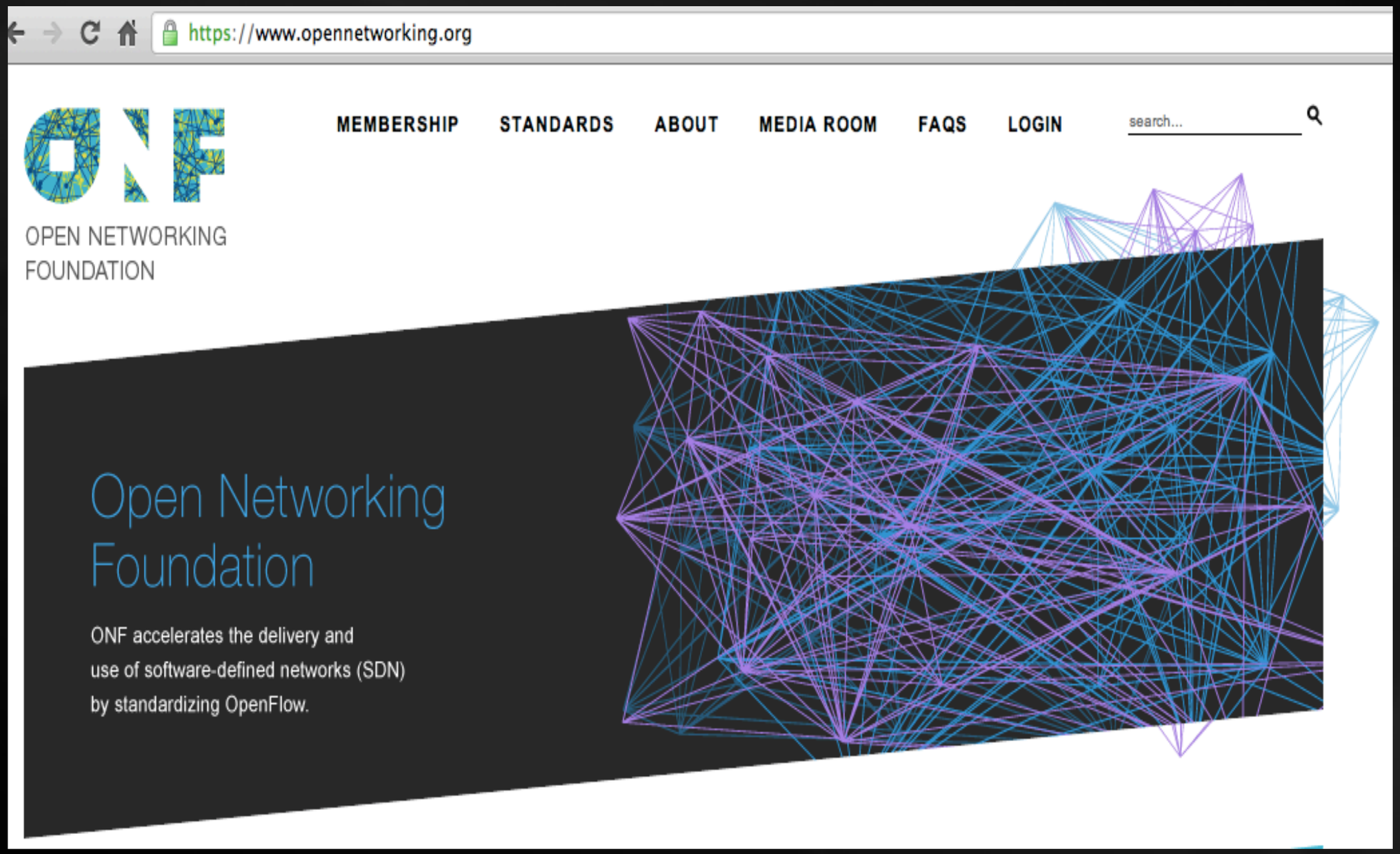
Where are they headed with networking?

Google

facebook



Get ready for Software Defined Networking (SDN)



What is Hardware Defined Networking?

An entire INDUSTRY organized along BOX function lines

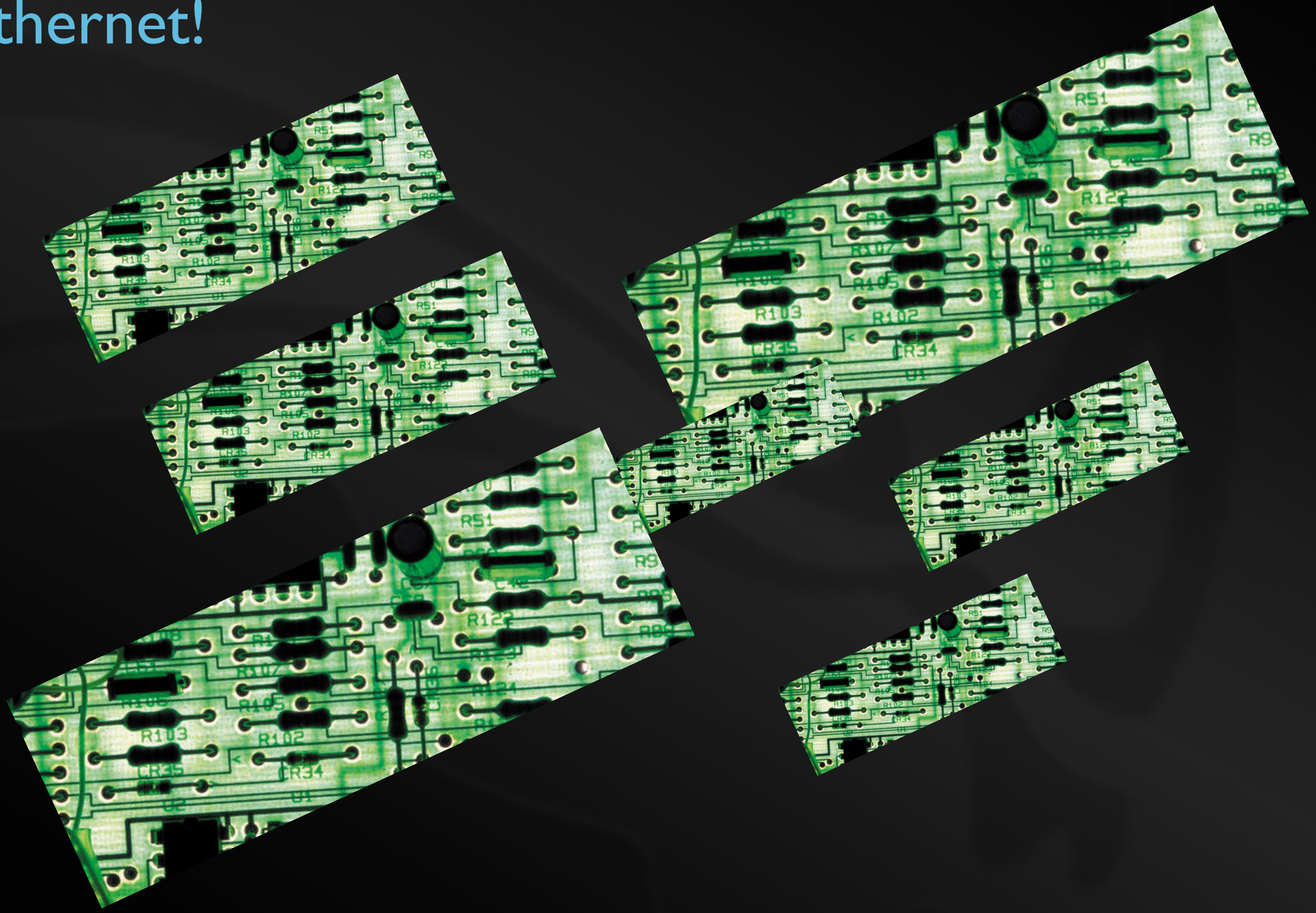
Routers and Switches

Firewalls

Load Balancers

WAN Optimizers

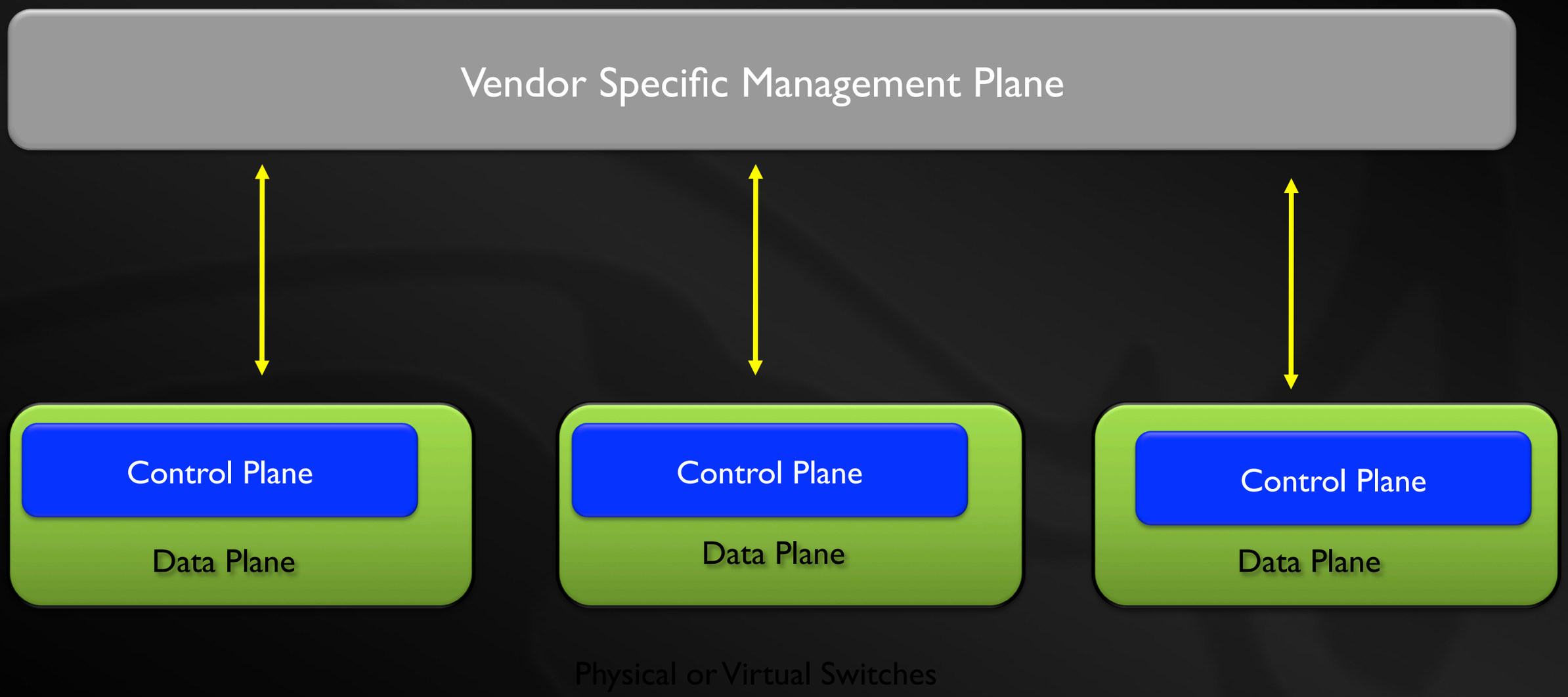
In reality, all these different “boxes” are the basically the same as servers: fast, cheap, and Ethernet!



However, consumers of networking solutions have not been enjoying the economics of fast cheap hardware like consumers of PCs, servers, tablets, mobile

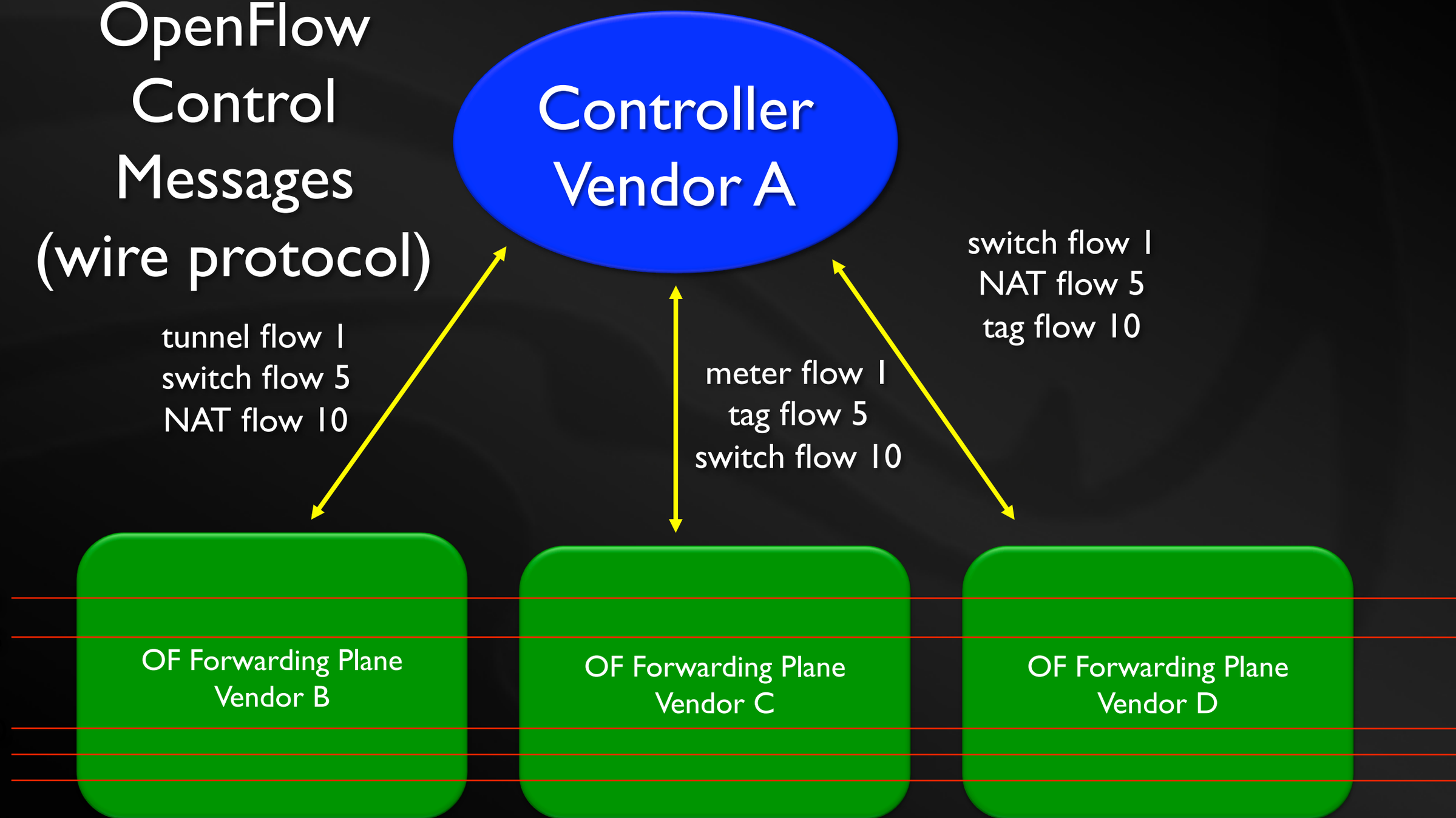
SDN is a "market correction" in the networking business where the bulk of value is finally transferring to software

Legacy Networking



No Application Awareness,
Labor Intensive to Deploy and Manage

A STANDARD Forwarding Plane and Logically Centralized Controller

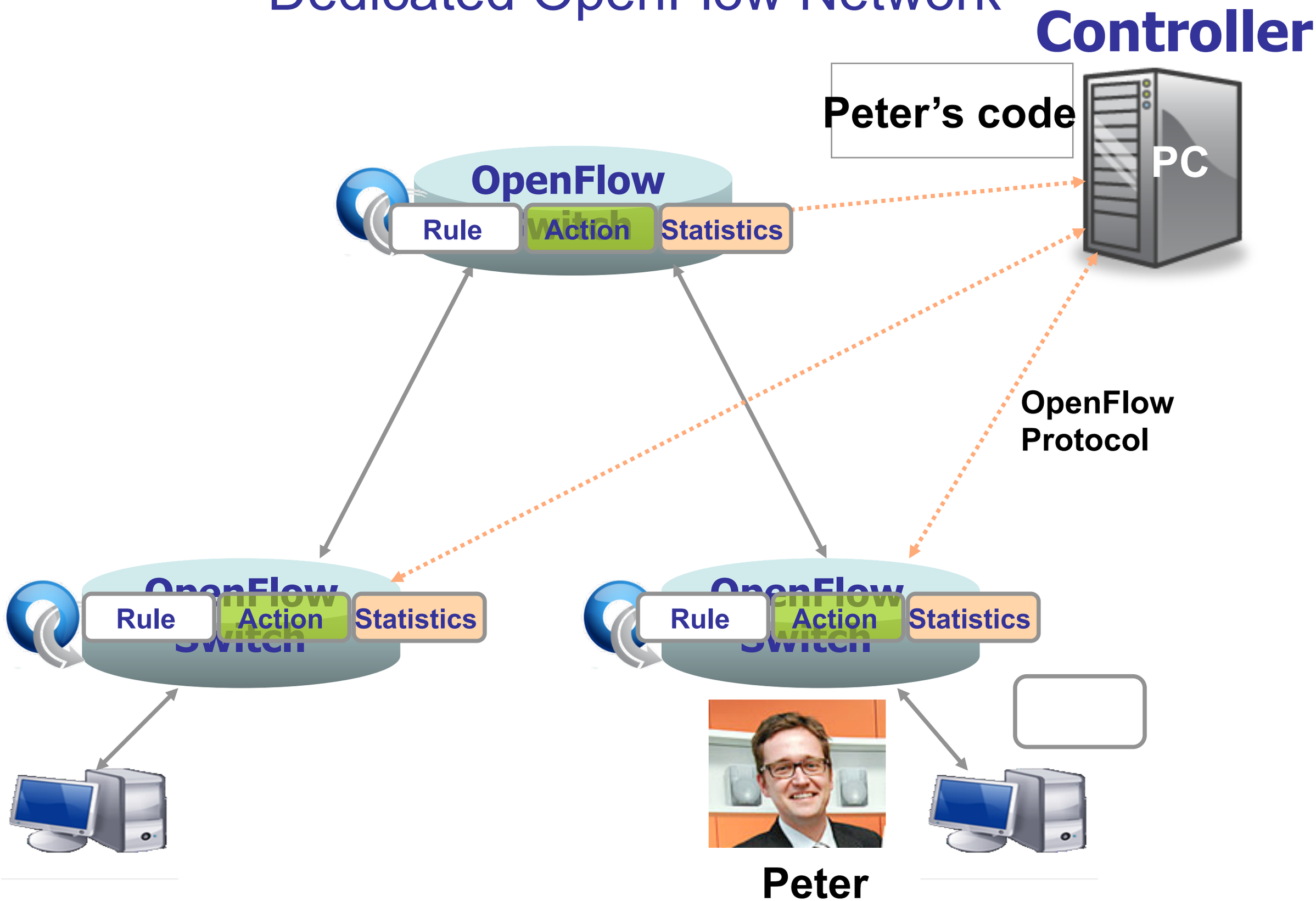


Capabilities across forwarding plane vendors are fairly uniform

Performance and capacity are primary differentiators

OpenFlow Usage

Dedicated OpenFlow Network



FlowForwarding.org

v20

Current

- Projects
  - LINC Switch – soft-switch implementing OF 1.2/1.3.1 and OF-Config 1.1
- Incubation
  - Hadoop Acceleration

Current Listed Contributors

Infoblox

Erlang SOLUTIONS

ARUBA networks

ERICSSON

Various Types of Open Source OpenFlow switches

1. Educational

   Goal: speed newcomers knowledge of OpenFlow technology and concepts

   Properties: easy to install/configure/use, source code easy to understand

2. Prototype

   Goal: a platform of easy implementation of new features

   Properties: source code easy to navigate with appropriate touch points

3. Bootstrap

   Goal: bootstrap viable products

   Properties: functionality, performance, fastpath optimizations

4. vSwitch

   Goal: software only switching

   Properties: optimized for virtualized and logical switching environments

The Four Major Open Source Projects (Snapshot as of Nov. 9, 2012)

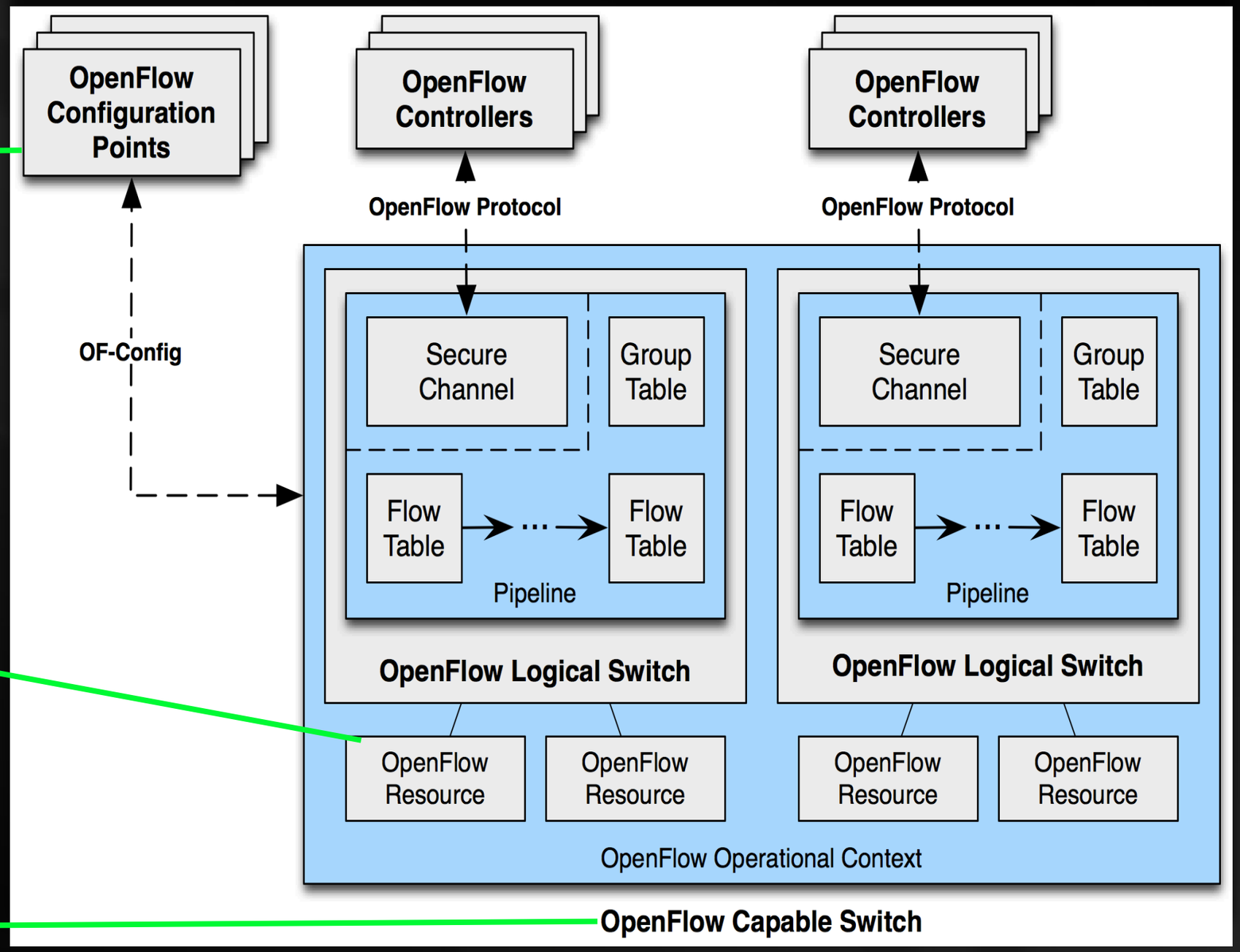
	OVS	Indigo2/LOXI	SoftSwitch	LINC
Bootstrapped by	VMWare	BigSwitch	Ericsson Research	Infoblox
Primary Type	vSwitch, bootstrap	bootstrap	educational, proto	proto, vSwitch
Fully Supports	OF 1.0		OF 1.0,1.1,1.2 (passed functional tests)	
Partially Supports		OF 1.0, 1.1, 1.2		OF 1.2
In Development	OF 1.1, 1.2		OF 1.3, OF-Config 1.1	OF 1.3, OF-Config 1.1
Optimal Platform	hypervisor	hardware fastpath	x86	multi-core x86
Language	C	C/Python	C	Erlang/ (C fastpath being scoped)
Strengths	performance, most widely adopted, tested on/supports lots of virtualization platforms	supports hardware fastpath, contains a static HAL (appropriate for static TTP)	easy to understand	takes advantage of multi-core, carrier grade features provided by Erlang/ OTP platform, concise code may mean easier management of TTPs
Modularity	organized for performance and virtualization	organized for hardware fastpaths	organized for understandability	organized for multi-core and a variety of fastpaths
Possible Markets	datacenter	hybrid	hybrid	distributed computing
Roadmap		OF-Config 1.1		fastpath in C, hardware fastpath

LINC Switch

- Partially supports: OpenFlow (OF) v1.2
- In development: 1.3.1, OF-Config 1.1
- Feature development focus:
  - Multi-core x86 and primarily OpenFlow networks
  - Less or no priority towards traditional networking interoperability
- Cross platform Implementation
  - Current: User space only
  - Future: support for Kernel space forwarding and hardware fastpath
- Carrier grade features available from Erlang/OTP Platform

LINC Switch Architecture

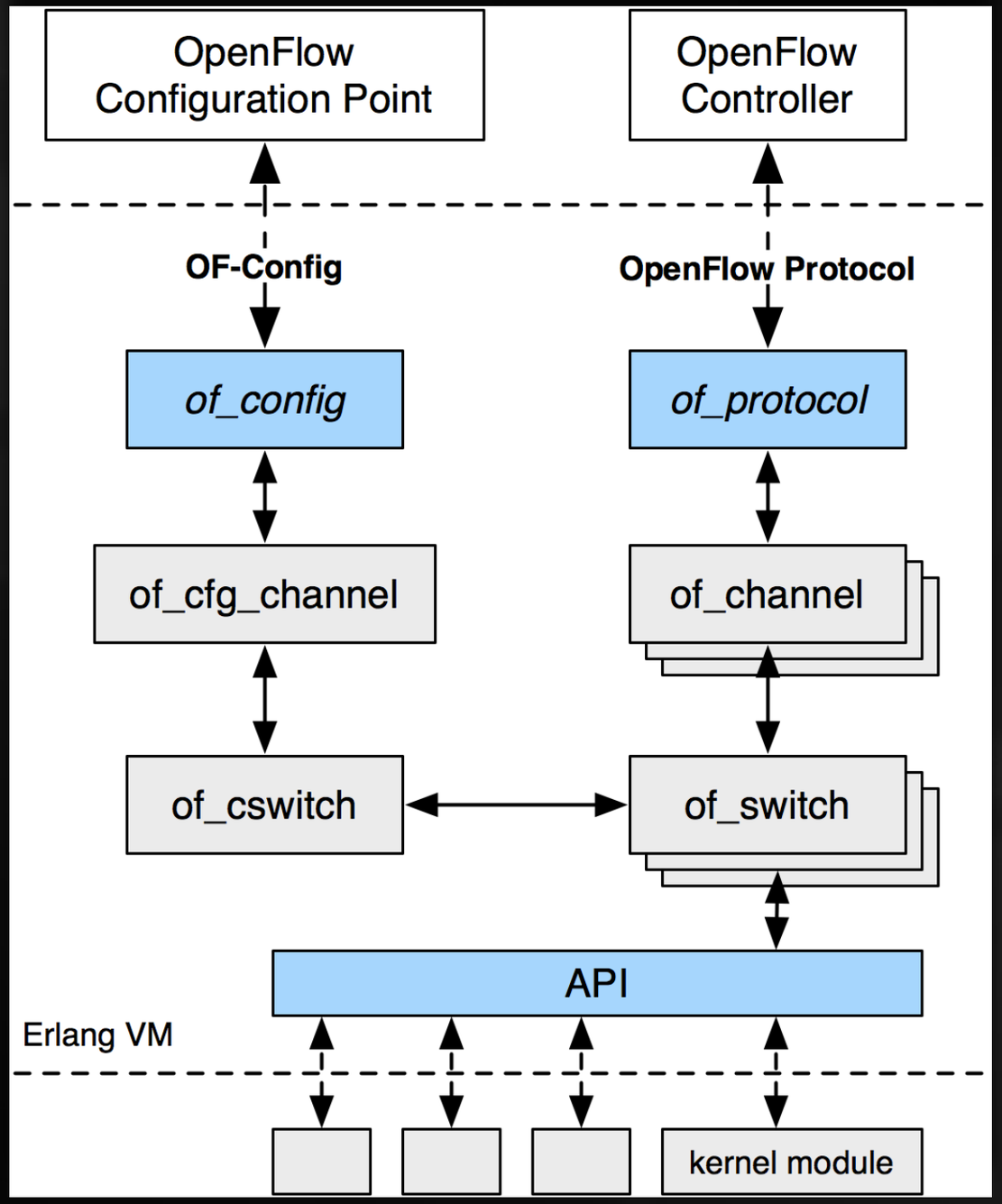
- Provisions OF Capable Switches
- Defines instances of logical switches
- Dispatches resources between switches



- Physical Ports

- Consists of ports & forwarding engine
- Container for multiple logical switches

Erlang Implementation Architecture



Why Erlang/OTP?

High Availability/ Reliability

- Built-in fault tolerance
- Software upgrade during runtime
- Suitable for server-side applications

Less Effort

- 4–20 times less code that C++/Java
- Suitable for rapid prototyping
- Powerful middleware and libraries

Scalability

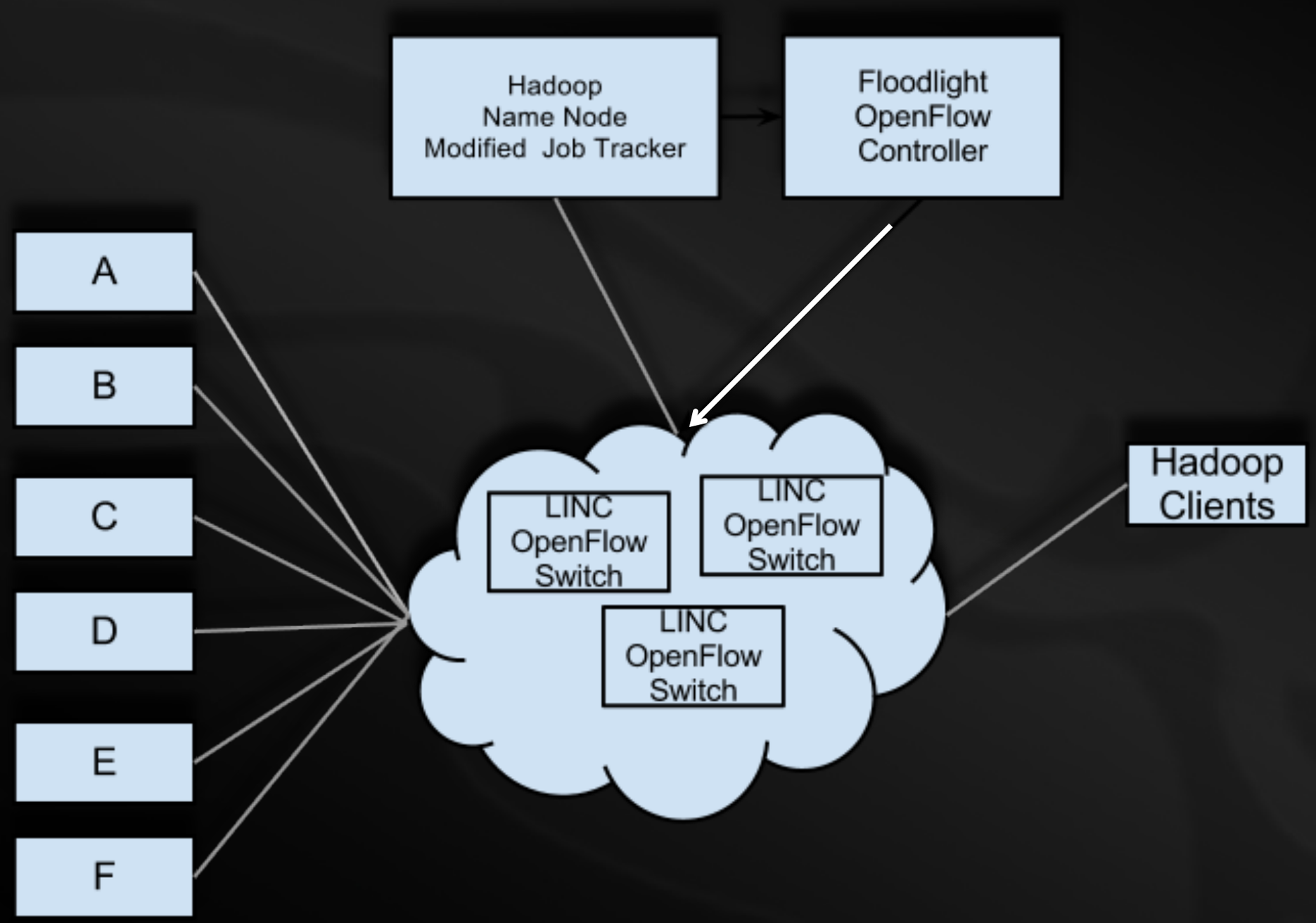
- Out-of-the-box Distributed Architectures
- Massive concurrency
- Symmetric Multi-core Support

THE BIT SYNTAX!

Hadoop Acceleration

- Leading open source compute cluster for BigData
- Designed to operate on commodity systems on commodity networks
- Supports MapReduce model of distributed computing
- Processes massive amounts of data for Analytics
- Clusters have very large data movement between execution phases of MapReduce
- Goal is to utilize OpenFlow to improve data transfer times by controlling the network from within the Hadoop MapReduce Framework
- New trends:
  - High Performance Computing is transitioning to use commodity clusters
  - High Performance Interconnects like Infiniband being explored for Hadoop clusters

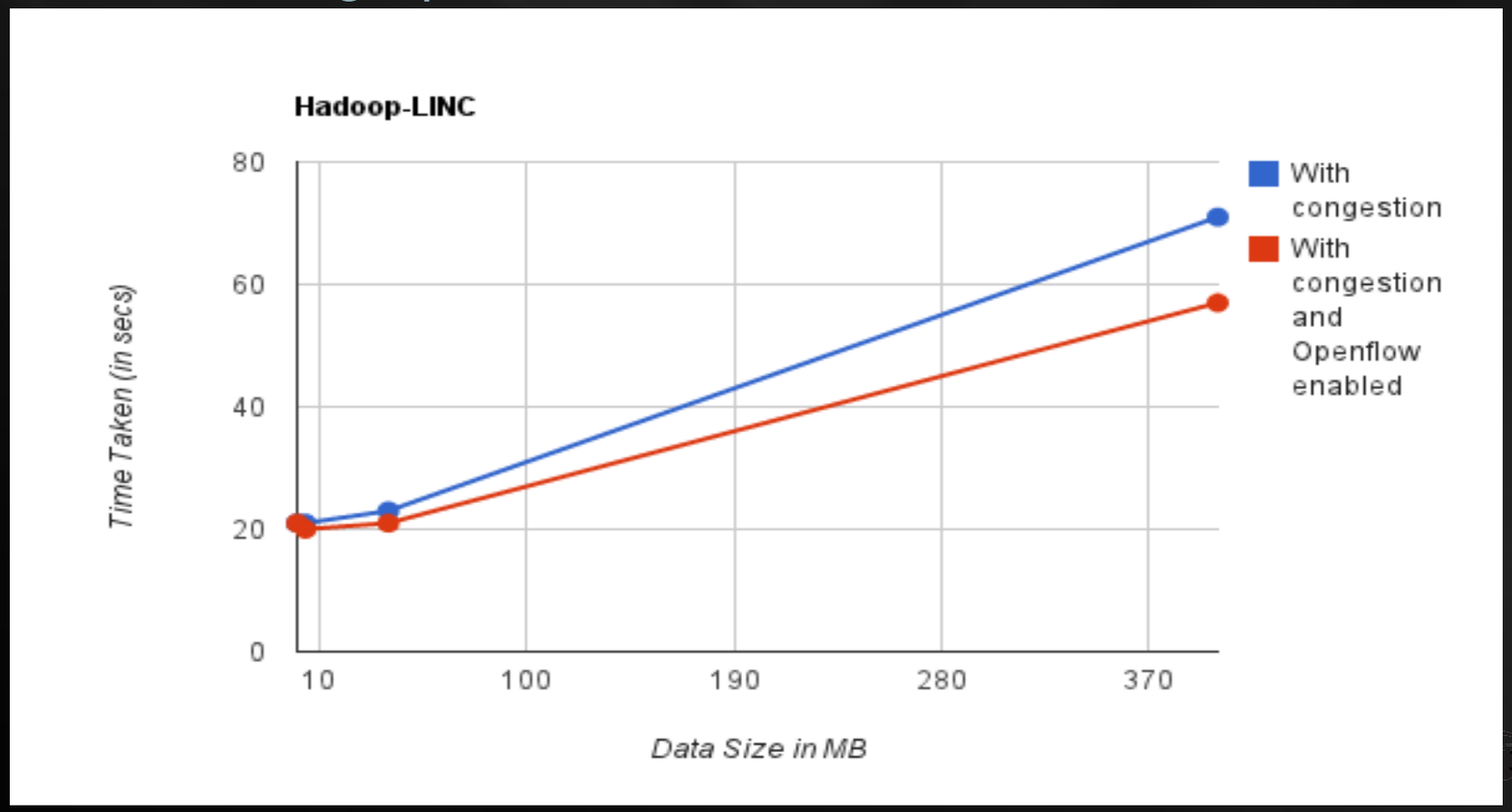
Hadoop OpenFlow Architecture



Hadoop Data Nodes
Modified Task Trackers

Hadoop Acceleration using OpenFlow

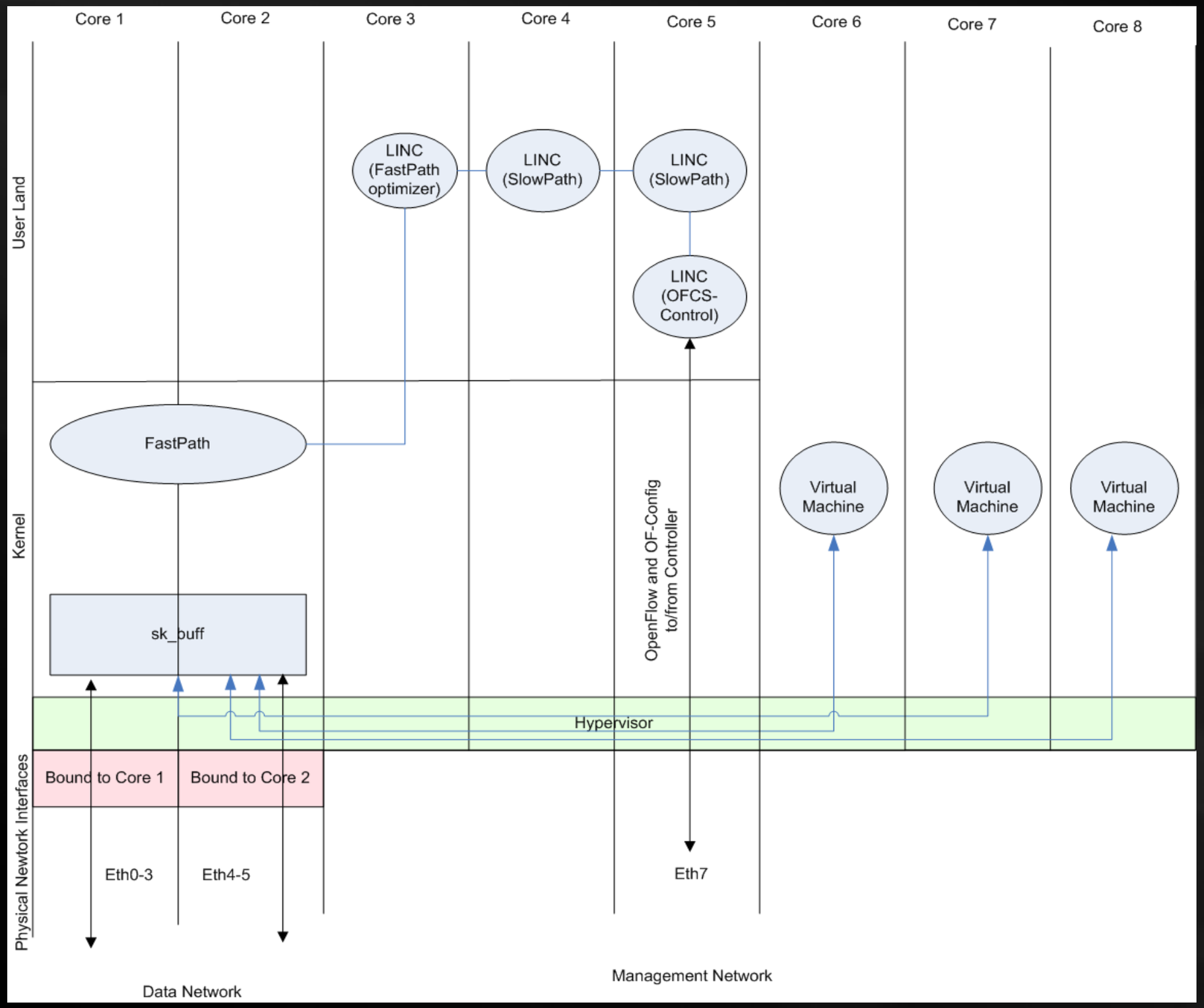
- Test program used: **Sort** from Hadoop benchmark (part of Hadoop distribution)
- Network Congestion created by **iPerf**
- Test run under two conditions
  - Setting lower priority for iPerf flow using OpenFlow QoS
  - Without setting priority



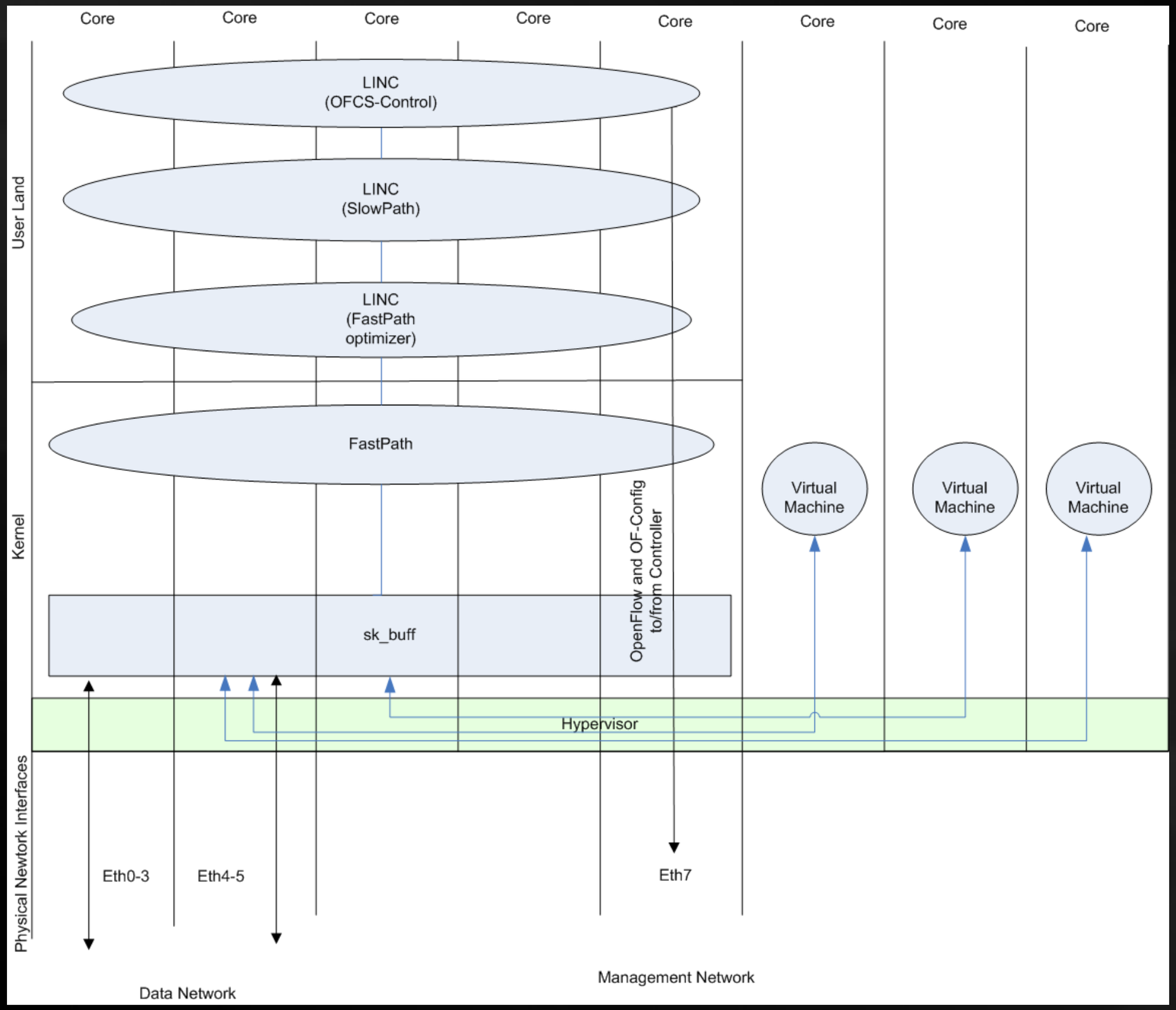
Possible LINC Roadmap Items

- Fast forwarding modules (C, Erlang on bare metal, Linux kernel, better networking hardware support)
- Performance with massive multi-core (64 cores and up)?
- How to build your own OpenFlow switch for less than $5,000
- Auto-provisioning/configuration with OF-Config
- Embedding into the OTP!!!

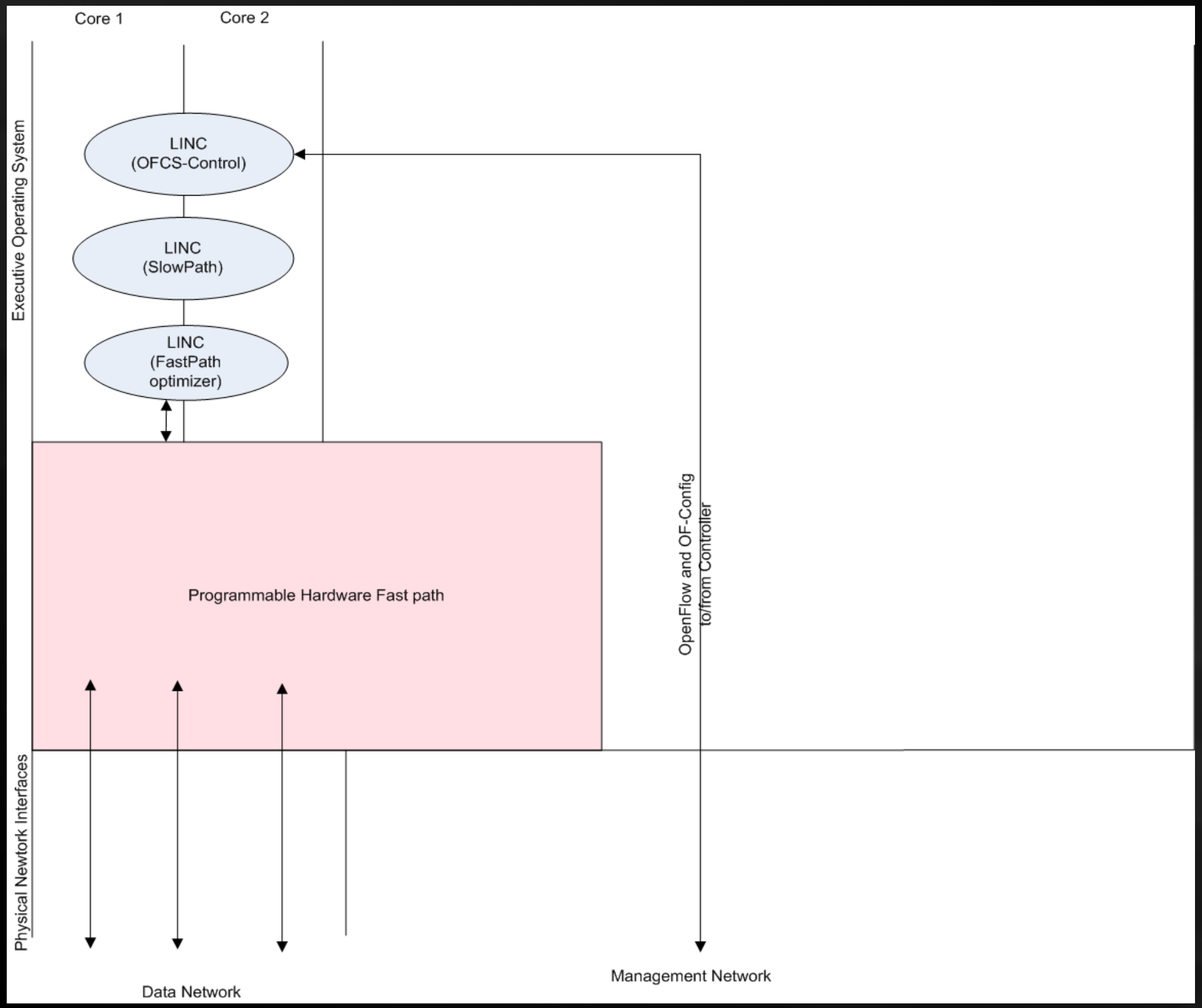
Fastpath with NIC-core pinning (e.g. DDIO)



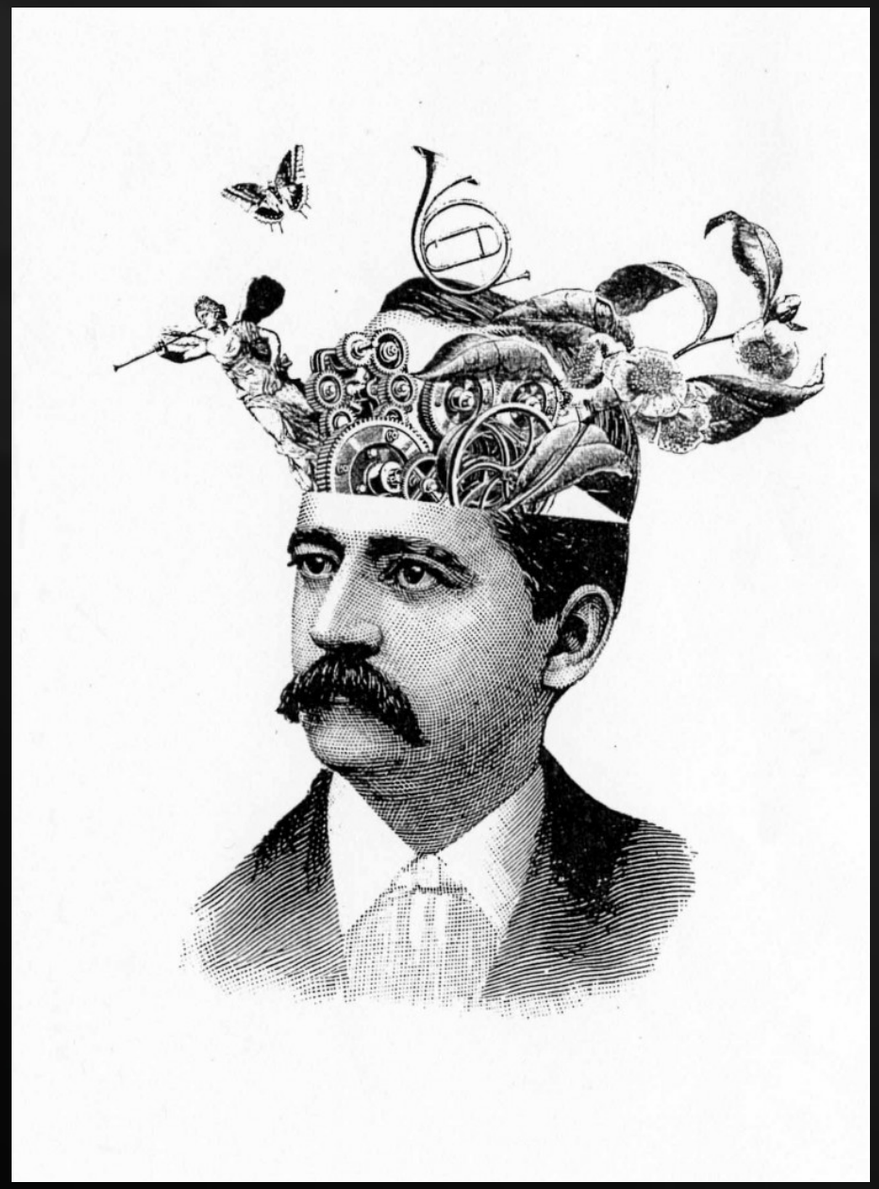
Fastpath using SMP kernel



Programmable Hardware Fastpath

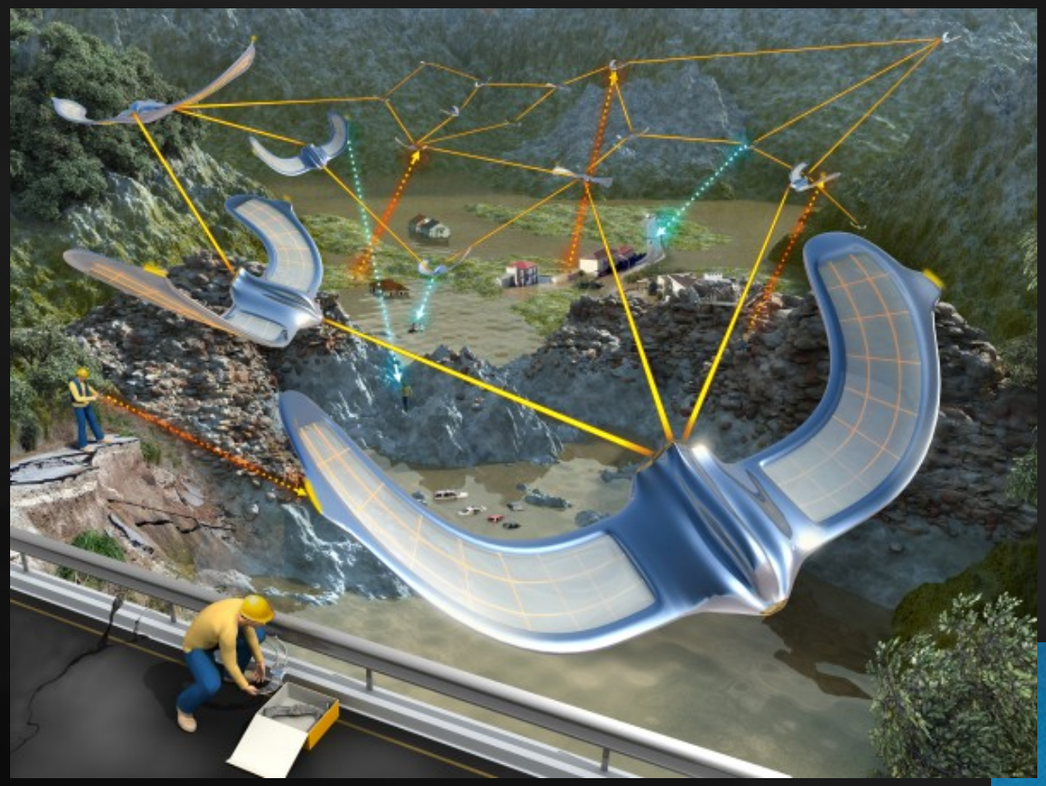


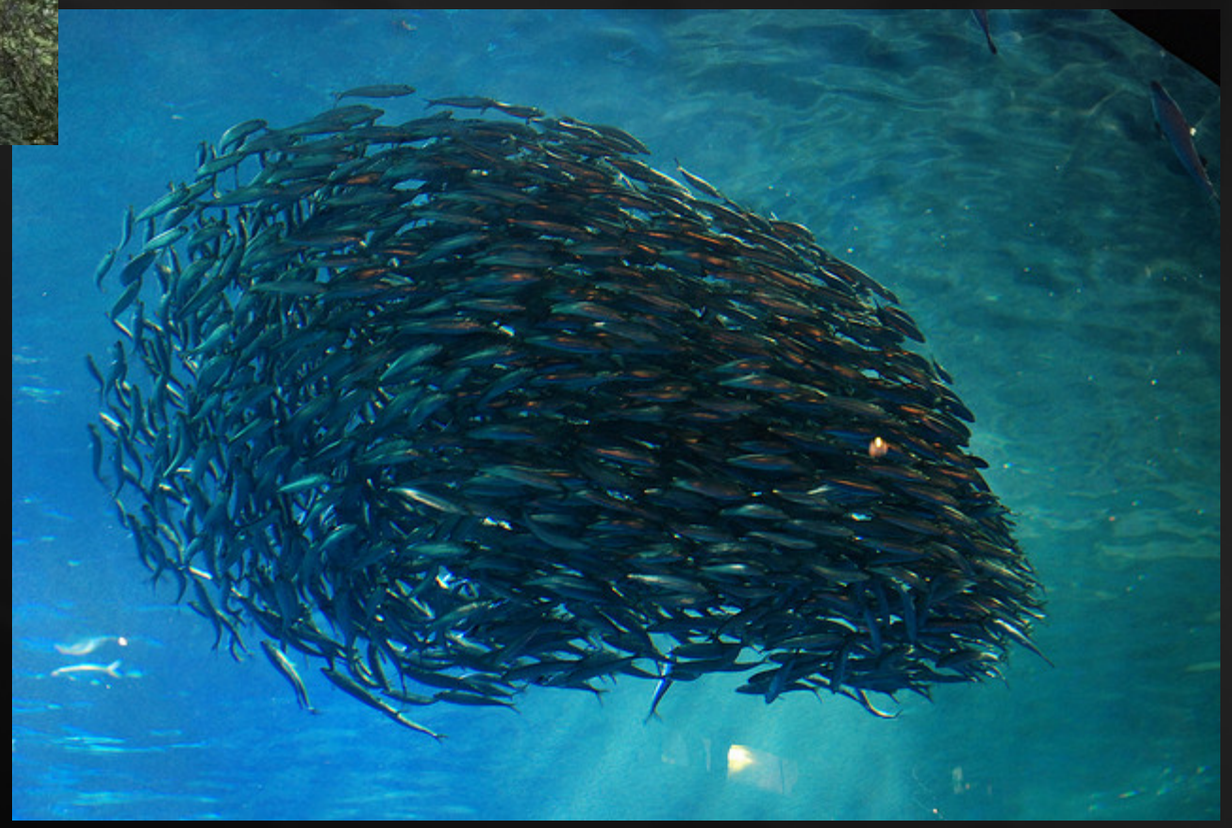
How can an Erlang/OTP developer program the entire network at runtime? (i.e. what abstractions, language extensions, mechanisms, etc.?)



For example, is I/O really a side effect? Is computation king?

Or is communication just as fundamental? Where are Joe's contract checkers??





Help us make it happen!

Email us @ info@FlowForwarding.org

http://www.FlowForwarding.org