Ethernet Alliance
Technology Exploration Forum 2013
"The Future of Ethernet"

Standardizing Software Defined Networking

October 16, 2013

This presentation has been developed within the Ethernet Alliance, and is intended to educate and promote the exchange of information. Opinions expressed during this presentation are the views of the presenters, and should not be considered the views or positions of the Ethernet Alliance

PANELISTS

- Curt Beckmann, Brocade
- Ed Crabbe, Google
- Bala Ramachandran, NetApp
- Michael Haugh, Ixia

Ethernet Alliance
Technology Exploration Forum 2013
"The Future of Ethernet"

Interoperable OpenFlow with NDMs and TTPs

Curt Beckmann
October 16, 2013

OpenFlow: A Control Language for Networks

- OpenFlow called "the x86 instruction set"
  - low level control of homogeneous switch capability
    - Highly desirable!? (Like PCs, right?)

- But a "uniform instruction set" is very challenging
  - Switches differ now,
    - Switches won't converge soon
    - Anyway, isn't competition at each layer an SDN promise?
  - Even PCs are diverse (32/64 bit, OS's)
  - And apps aren't coded in x86

- Let's recognize the diversity of network platforms
  - And notice that diversity has been handled before

Aspects of Network Device Diversity

- OF-Switch concedes feature diversity, kinda:
  - Lots of optional features
  - But with many options, what works with what?
  - Should app developers use optional features?
    - Or should they avoid them?
  - So x86, with "optional instructions"?
    - Really complicates interoperability
- Architecture diversity also matters
  - With single flow table, no problem
    - Flow entry fully defines what a flow is
    - And fully defines processing for the flow
    - It all fits in one message
  - But multiple flow tables is much harder…

Mapping low level instructions when pipelines differ

Table0
00 Count ← 16
01 Prod ← 0
02 Bit ← 1
03 If (RegX & Bit == 0) goto 05

Table1
04 Prod += RegY
05 Bit <<= 1
06 RegY <<= 1

Table2
07 Count -= 1
08 If (Count != 0) goto 03
[Or: If (Bit != 0) or (RegY != 0)]

Table3

00 Prod ← RegX * RegY

A smart compiler can see it's a "multiply"
As long as it can see the complete set of code

But if the code is in scattered in time?
If we ask the compiler to do the translation piecemeal, it becomes impossible

Similarly, mapping multi-table OF to legacy ASICs is tricky or worse… if we must do it **all** at run-time

But we actually don't have to do it ALL at run-time

And really, run-time is the wrong time

- Many variables affect SDN architecture
  - Apps, Controllers and Switches
  - Topology and Traffic
- Mapping multi-table OF is rather tricky, uncertain at run time
- Meanwhile, production operators NEED determinism, confidence
  - Typically they get it via testing of apps, controllers and switches in a few topologies and a variety of traffic loads
- With so much work done over and over prior to production run time… can't we "remember" what the app needed from the switches, and how pipelines were mapped?
  - Why redo it at run-time?

Historical footnote about production

- Google ran production networks on OpenFlow early, long before we heard about it…
- They saw many issues and conceived of new approaches
- They shared some of their ideas 2 years ago
  - Resulted in formation of Forwarding Abstractions WG, Aug 2012

What's the alternative?

- Multi-table OpenFlow changed the game
  - But the framework didn't change
  - Implicit assumption: same messages are enough
  - Valid only if all boxes offer complete OpenFlow pipeline
    - Because no mapping would be required in that case

- Bad news: hardware boxes don't (yet?) offer complete OF pipelines
  - Even new silicon will have diversity over time
  - And platform OS will vary
  - So networking will vary at least as much as PC's have varied

- Good news: we can change the assumptions
  - How do PCs handle diversity?

New assumptions → new perspective

- Instead of x86, we propose C or Java as the parallel
  - Both can be compiled for optimization
  - C is cool because it can be very low level
  - Java cool because it supports multiple models
    - "byte code" model for run-time portability, also compilable
- New framework: share switch pipeline "specs" before run-time
  - Comparable to picking the multiply instruction
  - Choose operands at run-time… that's enough
- To make it work, we must define pipelines in advance
- The pipeline is a "datapath model"

Defining Datapath Models in advance

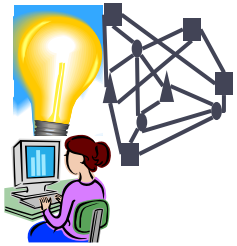
- "Datapath Model" must be detailed, unambiguous
  - Must spell out matches and actions allowed in each table
    - So no "pipeline surprises" at run time
- Apps will have different needs…no single datapath model will work
- So, a range of Datapath Models
  - Powerful platforms might support more than one model
  - Some apps may work on more than one model
  - Models need not be specified by ONF, others can do it too
- App and switch must agree on same model
  - A multi-vendor ecosystem means sharing → common language
  - "Agree" means synching up… "negotiation"
  - "Negotiable Datapath Model" → NDM
  - Must evolve over time as OF evolves

So, the new assumptions

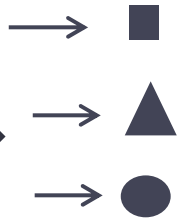
- Multiple unambiguous NDMs
- App / controller and switch must agree on NDM
  - Process for "agreement" defined by FAWG and CMWG
- NDMs based on, evolve with, OpenFlow architecture
  - 1st gen NDMs are OFS1.x-based: "Table Type Patterns" (TTPs)
- TTPs definable by ONF or ONF members
  - Using FAWG's common language for TTPs
- Anyone can define, so self-test scheme needed
  - Models have test info section for basic validation
  - 3rd party testing can go further
- Plus, a new framework!

NDM / TTP Lifecycle

1 Something drives need for new switch model



App provider has full solution idea



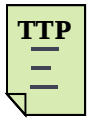
Drills down on specific element behaviors

ONF WG sees a common use case

Switch vendor shows key capabilities

2 Describe the model as a TTP

New



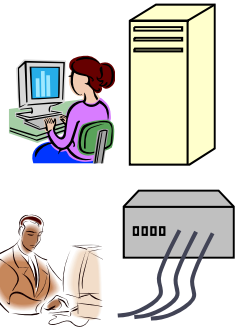
Describe switch behavior as precise subset of OF1.x model. Includes unique TTP identifier and version #.

3 Share the TTP Description

New

Share the TTP description with both sides (publicly, or under NDA)

4 Build support for TTP



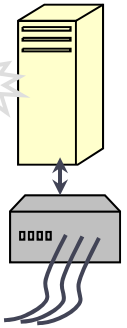
App provider and switch vendor independently add support for TTP in their products. Machine built switch plug-ins are a key goal.

5 Go to Market

Buyer considers product options (TTPs!), buys a solution and installs

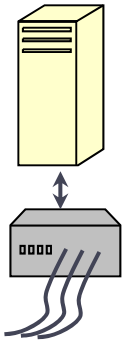
6 Connect & Pick TTP

New



App/ctrlr and switch check if TTPs supported, and if so they negotiate ID and parameters

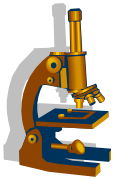
7 Same run-time msgs



App/ctrlr and switch go live! (flowmods, etc)

8 TTP-based testing

New?



Test labs will certify popular open TTPs

Benefits of TTPs

- Ease of development within a context of diversity
- Done such that interoperability is deterministic
- Interoperability visible to market participants
- No logjams required by “standardized profiles”
- Framework is for products that are “TTP aware”
  - Key for determinism when multiple flow tables needed
  - But TTPs also turned out quite useful for single tables!
    - TTPs can serve as precise test profiles
    - Can resolve the “optional feature” challenge
    - Visible to market participants

The Status of TTPs

- FAWG has documented the language of TTP Description
  - And we have shared near-final draft within ONF
- FAWG has described an interesting TTP in this language
  - Aimed at validating TTP language, framework
- ONF has a “working code” policy before releasing specs
  - TTPs are not “new protocol features”
  - So policy is “work-in-progress”
- Coming weeks:
  - Working code
  - Identify, define market-driven TTPs (you can help!)
  - TTP-oriented Tools
  - TTP FAQ

Technology Exploration Forum

Ethernet Alliance Technology Exploration Forum 2013 "The Future of Ethernet

IETF Related SDN Standardization

Edward Crabbe, Google
edc@google.com
October 16, 2013

The IETF & SDN

There are a number of standards efforts related to SDN under way in the IETF. Most prominently:

- PCEP
- I2RS

PCEP

Path Control Element Protocol

- MPLS focused
- Idea around for some time, PCEP as an RFC since 2009
  - Allows nodes (Path Computations Clients) to request TE operations from a controller (PCE)
  - Node owns LSP
  - Node dictates timing of requests
  - Controller has high level view of underlying network:
    - generally whatever exists in the IGP
  - mechanism for communication of IGP state to controller left unspecified
  - Controller does not have ‘demand level’ view

Stateful PCEP

The controller drives changes rather than the node. Specifically, the controller is capable of:

- dictating objectives and constraints
- dictating order of operations network wide
- initializing new ephemeral LSPs (on nodes supporting the capability)
- Use ofdifferent signaling mechanisms and encapsulations

The controller has ‘demand level’ visibility:

- Introduces concept of LSP-DB
- provides LSP level visibility to controller

I2RS

- Interface to the Routing System
  - Provide read/right access to device and protocol RIB states
  - Make use of existing protocols, recursion mechanisms, etc where applicable
- Relatively new effort. We're in the midst of:
  - Defining high level architecture
  - Protocol requirements
  - Info-models (RBNF at the moment)
  - Modeling language
- 'Model Based Networking'

I2RS Principles

- Designing from the ground up with extensibility in mind
  - Simple underlying transport
  - Relatively simple set of primitives required
    - Read/write model data
    - Trigger based subscription to data changes
  - Separate protocol from models
    - Explicitly Model Based
    - Devices may support subset of models
    - Models being designed to be readily extensible

What's Next

Stateful PCEP:

- Continue moving stateful PCEP and concomitant drafts towards RFC status in WG
- Develop Segment Routing extensions

I2RS:

- Select modeling language
- Design protocol or augment existing protocol
- Continue to develop models
  - RIB
  - BGP
  - Topo

Technology Exploration Forum

Software-Defined Storage & Standards Implications

Bala Ramachandran

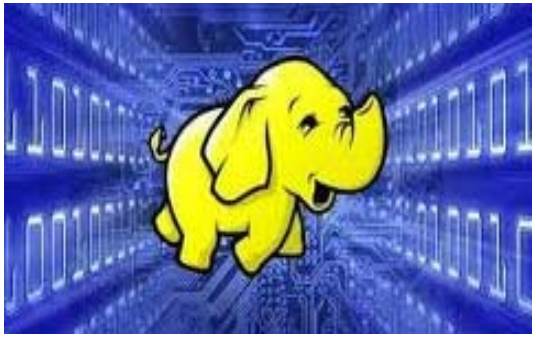
Sr. Manager, Product Strategy
Office of CTO, NetApp

Acknowledgements

- Gokul Soundararajan
  - Member of Technical Staff, Advanced Tech Group, NetApp

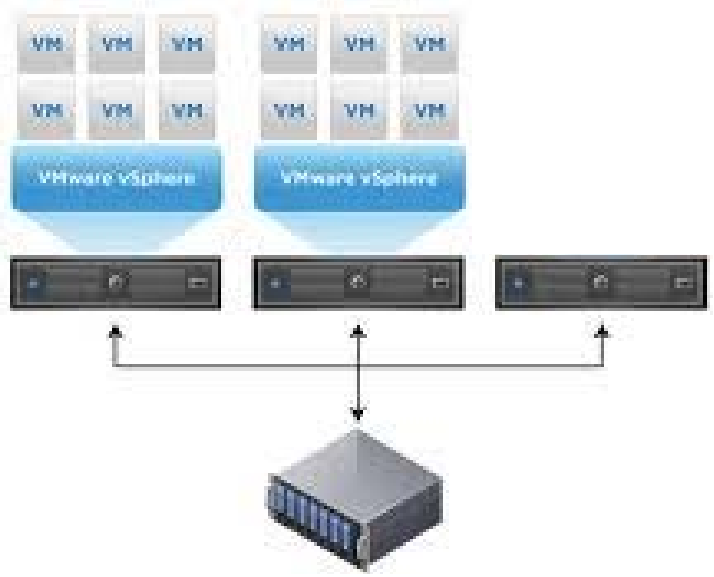
- Kaladhar Voruganti
  - Senior Technical Director, Advanced Tech Group, NetApp

Mega-Trends Shaping Storage Industry



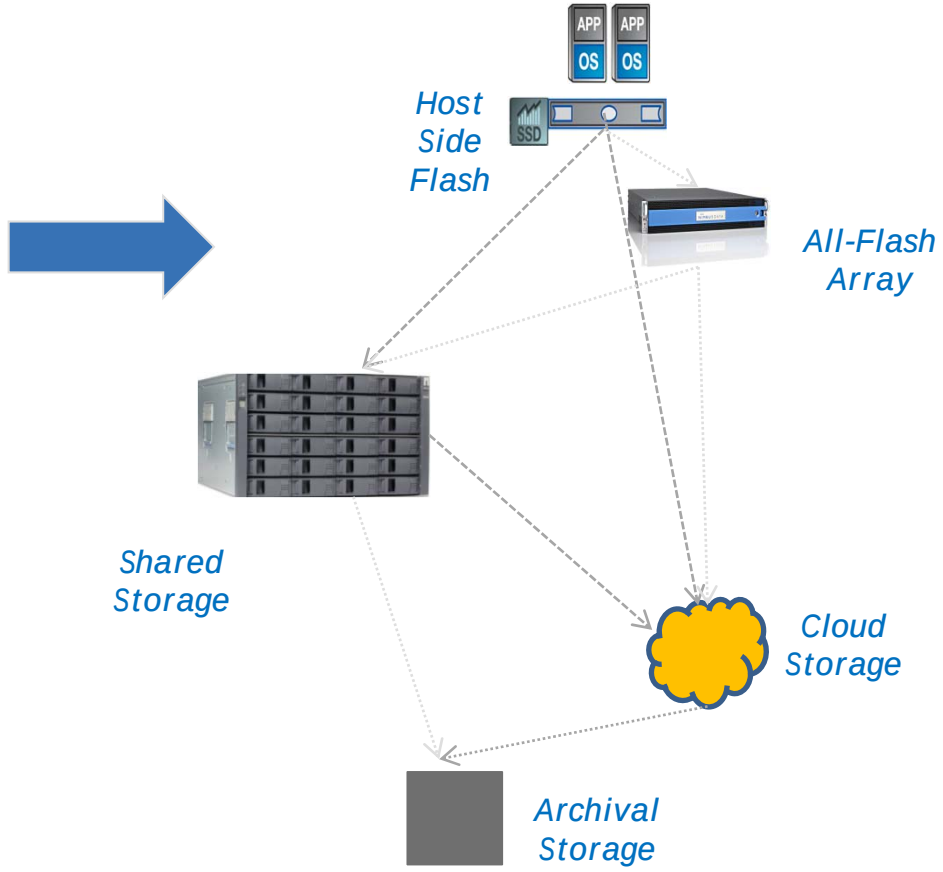
Evolving Storage Architecture

Traditional



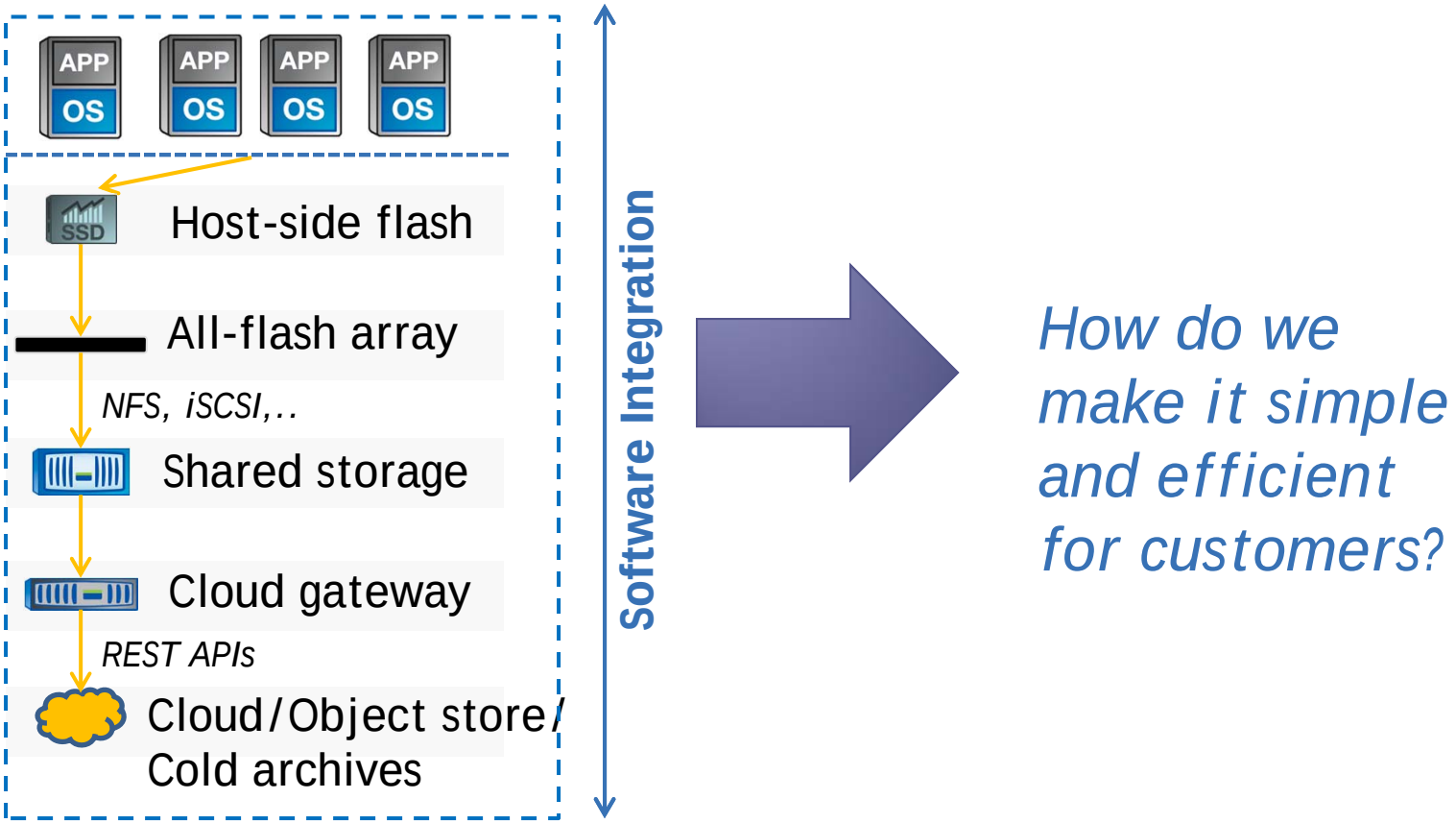
NAS/SAN Shared Storage

Emerging



What does this mean to storage?

Virtualized STACK of dynamic, heterogeneous resources



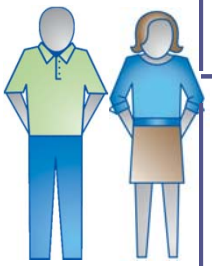
How do we make it simple and efficient for customers?

Management by SLOs

CIO

"Run Exchange for 5000 users"

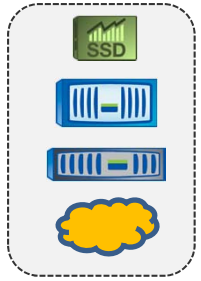
App Admins



Compute

Network

Storage



Performance
- IOPS

Availability
- Annual Downtime

Reliability
- RPO/RTO

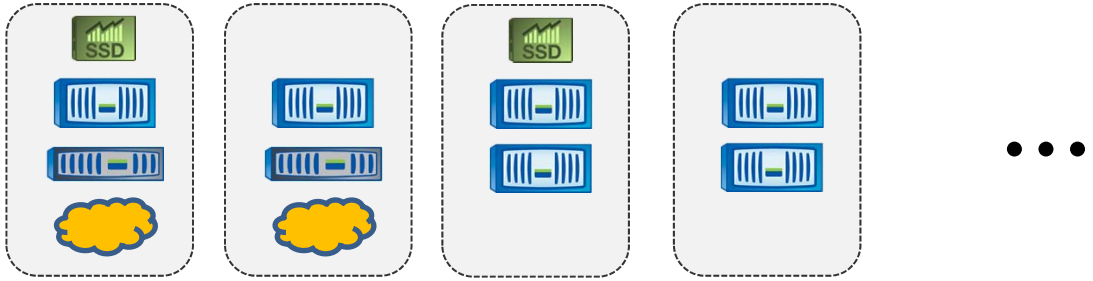
Security
- Encryption levels

Space Guarantee
- Thin Prov – yes/no

SDS - Simplified

Essence:-

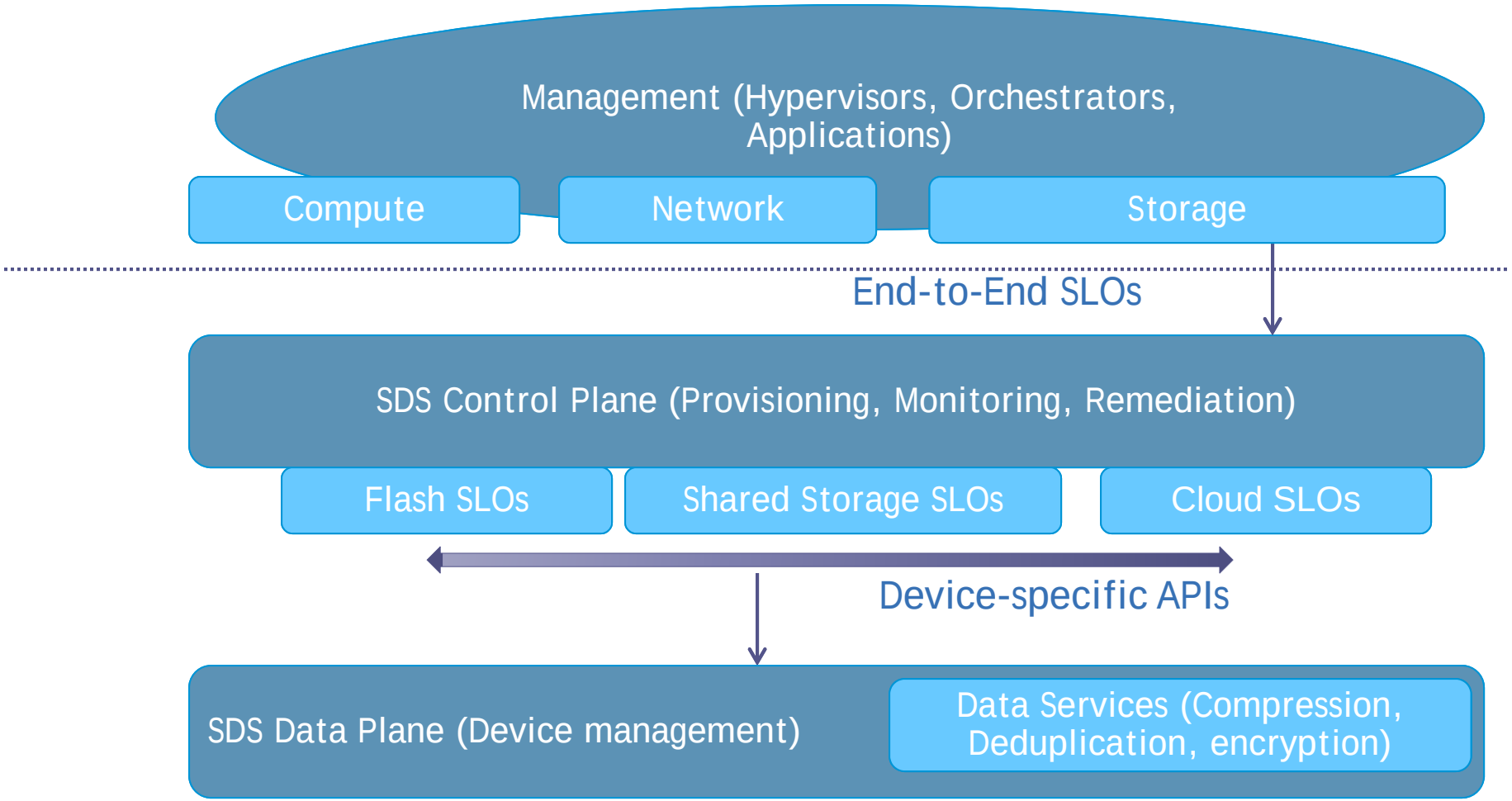
Which of these is the most efficient stack for each workload?



Do this in a way that is

- Simple/Automated
- Dynamically coordinated end-to-end
- Programmatic
- Unified Management

SDS High-level View



Value Proposition of SDS Model

- Application/Business-Aware Infrastructure
  - Dynamic usage patterns
  - App-specific policies
  - Workload and data mobility
- Automation and orchestration in heterogeneous environment (solves a hard problem for the customer)
- Operational efficiency

SDS Adoption Drivers

- Change in buying centers to enterprise application groups
- Increasing adoption of Unified Cloud Management Platforms eg. Openstack
- Popularity of Open source solutions
  - Programmability with Open APIs
- SIs and Resellers offering SDS “solutions” on heterogeneous hardware partnering with orchestration vendors

SDS Adoption Inhibitors

- Vendors need to evolve to flexible software business models
- Need for unified standards for APIs
  - Could be difficult to achieve because of weakening of vendor lockin
  - Need for vendors co-operation
- Customers need to realize expected cost savings
  - Dependent on vendor go-to-market strategy

Standardization

TOSCA:
Topology and Orchestration
Specification
for Cloud Applications

Conclusion

- Software Defined Storage offers a lot of technological promises
- Adoption will depend on how it gets translated to economic value for customers
- Standardization is very critical for broader adoption of this technology

Technology Exploration Forum

Ethernet Alliance Technology Exploration Forum 2013 "The Future of Ethernet

Standardizing SDN and OpenFlow through Testing

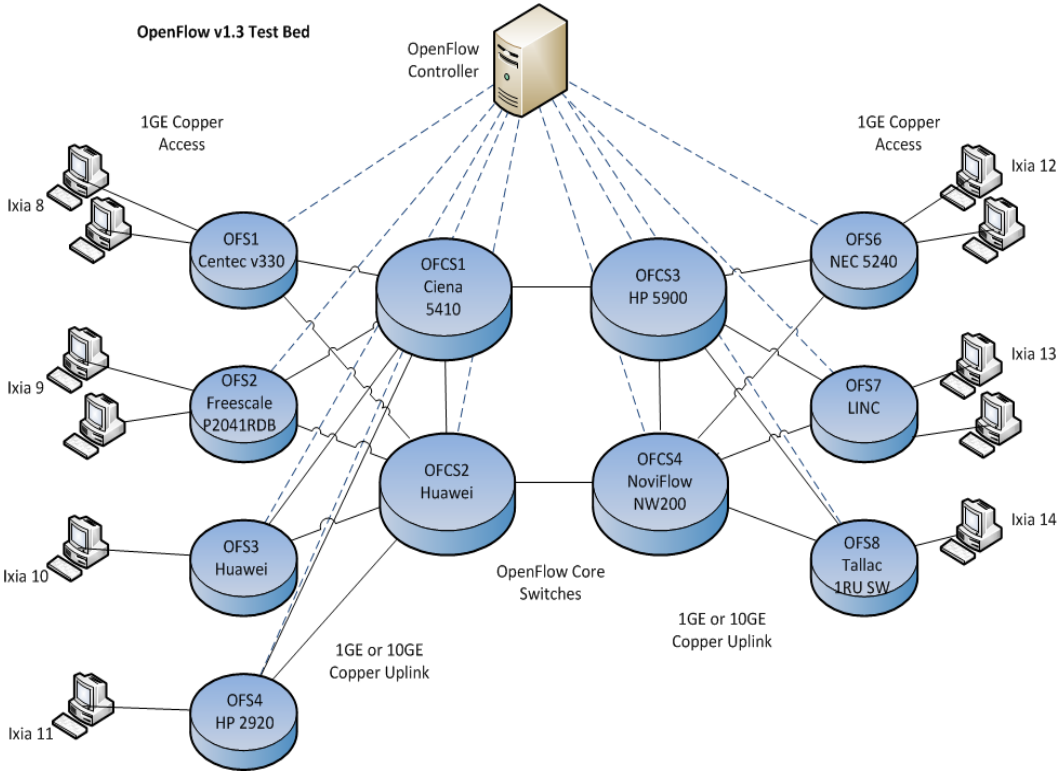
Michael Haugh, Ixia
mhaugh@ixiacom.com
October 16, 2013

Testing OpenFlow – Testing & Interop WG @ ONF

- Testing & Interop WG created in 2011
- Facilitating testing through interoperability "plugfest" events
- Conformance program for OpenFlow switches
- Defining benchmarking methodologies
- Drafting papers on testing and interoperability issues

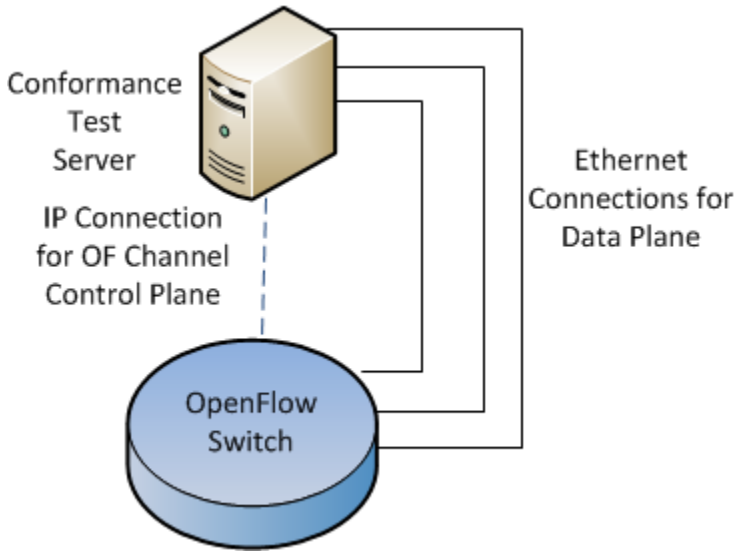
OpenFlow Interoperability Testing

- 2 Events in 2012, mainly focused on v1.0
- June 2013 event
  - OpenFlow v1.0: 6 controllers, 13 Switches
  - OpenFlow v1.2: 2 controllers, 1 switch
  - OpenFlow v1.3: 9 controllers, 11 switches
- Example Applications
  - LLDP based Discovery
  - Layer 2 VLAN based path provisioning
  - MAC based learning/forwarding
  - Hybrid - Routing over OpenFlow
  - MPLS based forwarding
  - Load balancing/ECMP
  - Failover scenarios with multiple controllers or clusters
  - IPv6 based learning and forwarding
- Next event November 4 – 8th



OpenFlow Conformance Testing

- **OpenFlow v1.0 Switch Certification Program**
  - 200+ test cases
  - GA certification launched
  - First vendor certified
- **OpenFlow v1.3 Switch Certification Program Underway**
  - 750+ test cases defined
  - Expected in 1H 2014
- **Three Lab Partners**
  - InCNTRE – Indiana University
  - BII, Beijing China
  - UNH-IOL, New Hampshire



Challenges and Work Ahead

- OpenFlow v1.3 defines 40 Match fields, 13 mandatory 27 optional – creates challenges for testing and certification – increasing going forward
- Vertically integrated systems (App/Controller/Switch) have best success in testing – TTPs could help to define application requirements for switches
- ONF is moving quickly, v1.4 out, v1.5 starting
- "Carrier Grade SDN" is now becoming a hot topic with carriers and performance metrics – scale, resiliency, security and other features to deliver SLAs are important!
- Missing features (OAM, BFD, SNMP) are increasing mix of hybrid deployments
- SDN/OpenFlow is expanding into many areas – Optical, Wireles/wifi, NFV and others

Technology Exploration Forum