Cloud/IaaS Platforms

I/O Virtualization and Scheduling

Dave Cohen, EMC
USENIX LISA 2013
Invited Talk

Disclaimer

The contents of this presentation is derived from publicly available sources

The slides presented here are a technical survey of the architecture common to Cloud and HPC platforms. It is intentionally Vendor-Neutral. The slides and my presentation are not statements from my employer and are not intended to present my employer's position or strategy, nor is this presentation an assessment of products from other vendors, their position or strategy.

All copyrights used in this presentation are owned by their respective companies or individuals.

Agenda

- Introduction
- Part I - From Desktops to Mobile — The shift from Inside-Out to Outside-In, the corresponding explosion in data growth, and how to address this.
- Part II - Virtualizing the I/O stack — Dealing with the rise of multi-core/multi-thread, horizontally-scaled applications and the resulting impedance mismatch between compute and storage.
- Part III - IP Internetworking — Network Virtualization as the means for supporting I/O Forwarding
- Part IV - Project Autovias — Network Partitioning through virtualization.
- Conclusion

Abstract

Cloud-based, Infrastructure-as-a-Service (IaaS) platforms present a simple data center resource model to their end-users. The end-user provisions logical resource units of compute, network, and block storage. They then attach a compute unit to one or more network and block storage units. This “attach” and subsequent “detach” of emulated network and storage devices embodies what is referred to as I/O virtualization, the act of scheduling I/O resources independent of compute. For this session we will outline this I/O virtualization concept and then dig deeper into how Cloud/IaaS platforms leverage it for both local and global/data-center-wide scheduling.

About Me

Dave is a strategist and researcher in EMC's Office of the CTO where he focuses on the costs, scalability, and performance of data center infrastructure, especially in a Cloud setting. During his tenure at EMC, he has provided technical leadership in the areas of Network Virtualization, OpenStack, and converged infrastructure platforms such as OpenCompute. Prior to joining EMC, Dave spent ten years on Wall St working on various facets of data center infrastructure for companies such as Goldman Sachs and Merrill Lynch.

Agenda

- Introduction

• Part I -	From Desktops to Mobile	The shift from Inside-Out to Outside-In, the corresponding explosion in data growth, and how to address this.
• Part II -	Virtualizing the I/O stack	Dealing with the rise of multi-core/multi-thread, horizontally-scaled applications and the resulting impedance mismatch between compute and storage.
• Part III -	IP Internetworking	Network Virtualization as the means for supporting I/O Forwarding
• Part IV -	Project Autovias	Network Partitioning through virtualization.

- Conclusion

Transitioning from the "Inside/Out" to the "Outside/In"

A Firm's view of the world:

1. Inside/Out view
   - Employees at their desks in Firm office facilities, using terminals to access data services, operating in Firm data center facilities
   - Customer Relationship Management (CRM) packaged software such as Oracle's (formerly Siebel) CRM product
   - Parasitic, "In-House" application software (e.g. Monthly Reports, Data Loads, etc), each application allocated its own infrastructure
   - Pair-wise connectivity: between Office and Data Center, between Data Centers
2. Outside/In view
   - Employees using smart phones, tablets, laptops to access data services, operating in Firm data center facilities as well as facilities operated by 3rd parties
   - Software-as-a-Service (SaaS) based CRM offerings such as SalesForce.com
   - Platform-as-a-Service (PaaS) based, parasitic applications (e.g. Monthly Reports, Data Loads, etc), the collections of applications sharing the PaaS platform's services.
   - Internet/Cloud, Anywhere/Anytime connectivity

The Shift from Inside/Out to Outside/In makes it increasingly attractive to transform Data Centers into a Shared Resource. This, in turn, shifts the operator's focus to Economies-of-Scale.

How does a Data Center operator realize Economies-of-Scale in an Outside/In economy?

1. Economies of Scale - desirable characteristic of a Plant/Factory where factors cause the average cost of producing something to fall as the volume of its output increases.
2. The Data Center as a Computing Plant/Factory - the Plant/Factory is a Data Center (or set of Data Centers) whose processes transform electricity into mathematical work and waste heat.
3. Operational-Cost-per-Megawatt-of-Power - an average rate that consists of: networking, power and cooling equipment, plus staff; the rate is expected to decrease as the size of the data center increases.
4. Bringing the Operational-Cost-per-Megawatt-of-Power down via Sharing and Automation
   - Sharing - the networking infrastructure plus power and cooling equipment
     - Leveraging the Power/Cooling infrastructure across a larger environment affords numerous efficiencies
     - Collapsing Application, Storage, Backup, and Management Networks on to a single "wire"
   - Automation – increase the productivity of operational staff through increasing levels of automation

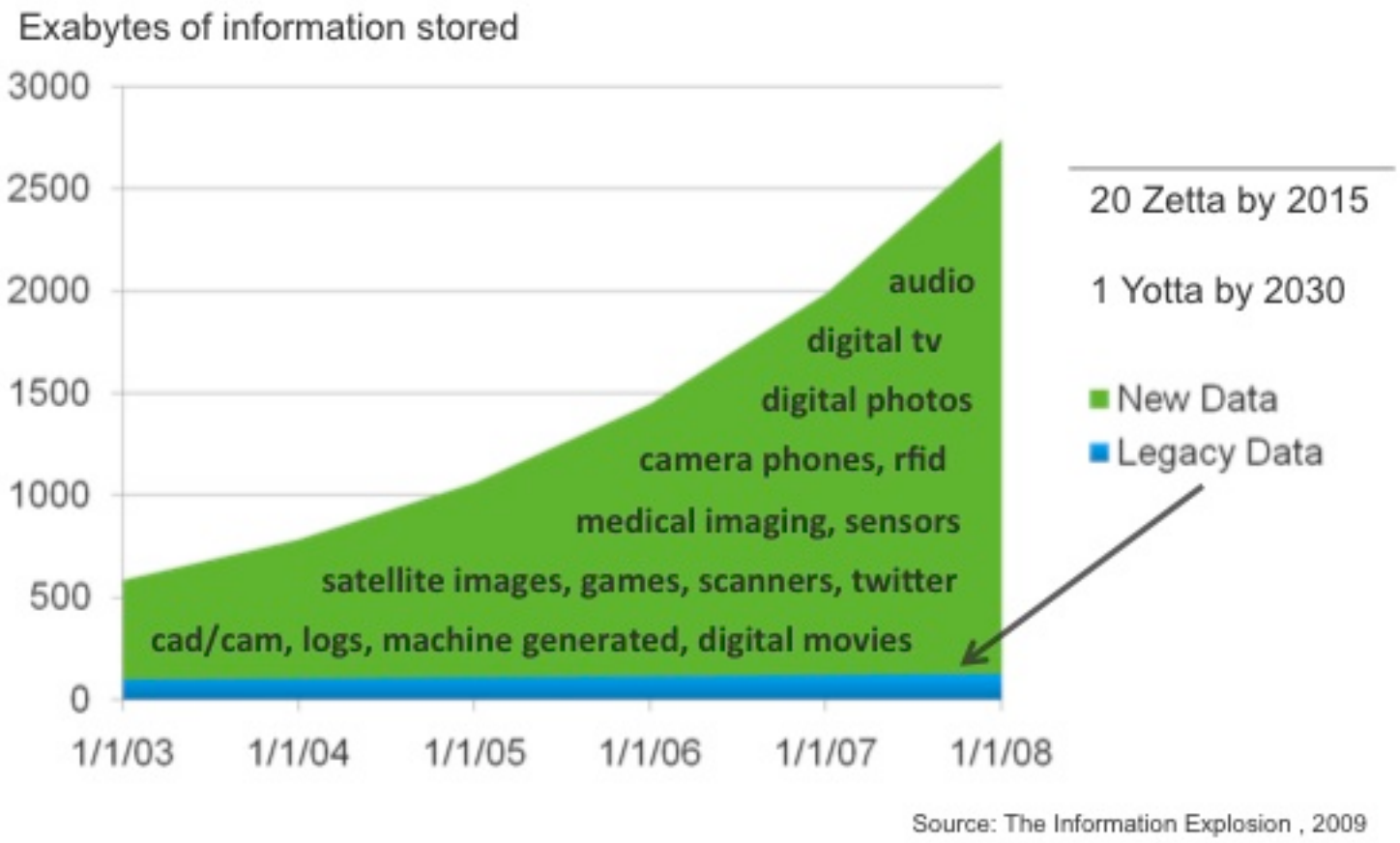
For the purposes of this talk, Sharing the Physical Network Infrastructure and Increasing the levels of Automation are key to realizing Economies-of-Scale.

Challenge: Infinite Data Growth and why Storage needs to be treated differently

1. Explosion of Data Sources in the Outside/In World - digital pictures and videos produced by Smart Phones and Tablets, Ultrasound and MRI machines, Genomic Sequencing, Automobiles, Homes, etc.
2. Data is Always Available - from the user's perspective, the expectation is that, once uploaded to the data center, this data is stored forever.
3. The Shelf-Life of Data - the reality is that across all workloads, data has a shelf-life. From its inception, data is accessed with some frequency over a finite period. After this initial period, the frequency drops off rapidly. Finally, at some point the probability of access becomes negligible.
4. Data Redundancy - In the Inside/Out world, data was stored using RAID for protection and mirrored to a second site for recoverability. Conservatively, this meant data had a redundancy factor of 2.x or greater. Worse, only a small fraction of the overall data-under-management was being used in any 24-hour period.
5. Reducing Redundancy - as the size of data-under-management grows beyond some level (eg 10s of petabytes); bringing the redundancy factor down below 2 becomes increasingly attractive.

Unbridled growth in the size and diversity of data-under-management has forced data center operators to look for ways to decrease the data redundancy rate.

Explosion of Data



Source: The Information Explosion , 2009

Reducing the Data Redundancy Rate through Multi-Site Erasure Coding

1. Reducing the Data Redundancy rate- In the Outside/In world, multi-site erasure coding has/is being adopted so that data center operators can bring the redundancy factor down below 2x.

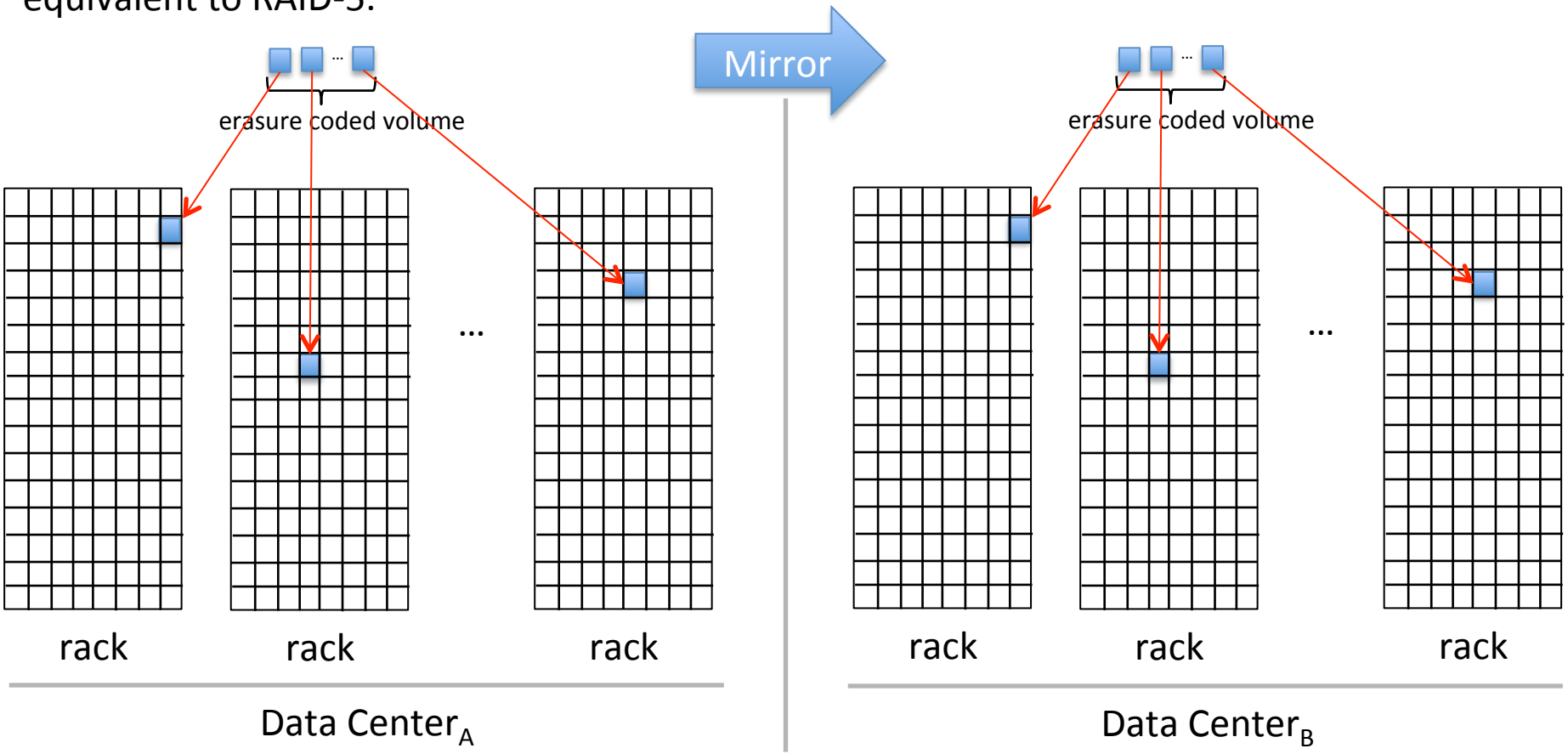
2. Scale - use of multi-site erasure coding is predicated on a minimum of 3 interconnected data centers where the connectivity provides high-bandwidth and good latency between the sites.

3. Caching - in addition, retention policies increase the redundancy factor so that erasure-coded data is not accessed. Instead, data with a high-probability of access has an additional copy retained online via snapshots. As the probability of access decreases this additional copy is deleted.

Adoption of Multi-Site Erasure Coding is predicated on Scale and Caching; the latter hiding issues with performance and access-times.

Distributed Storage with Dual-Site Mirroring

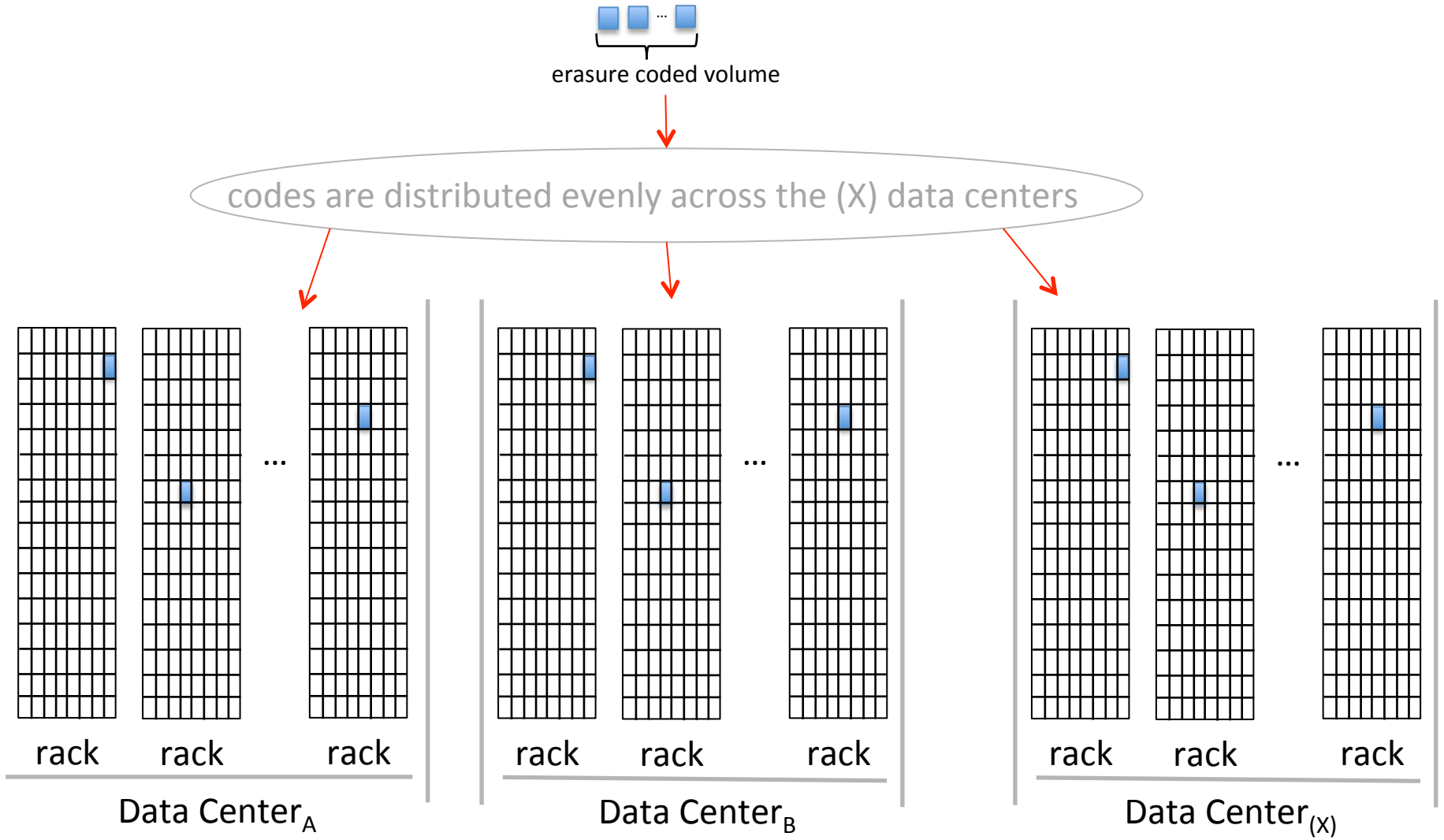
This approach is attractive for organizations who have data centers that were paired for mirroring. The erasure coding improves reliability and availability with redundancy roughly equivalent to RAID-5.



Assumes additional Local Reconstruction Codes (LRC) support (or equivalent), there is a high-probability that a volume can be recovered locally.

Distributed Storage with Multi-Site Erasure Codes

Improves on Reliability and Availability of Dual-Site while reducing redundancy below 2x.



Assumes additional Local Reconstruction Codes (LRC) support (or equivalent), there is a high-probability that a volume can be recovered locally.

Agenda

- Introduction
- Part I - From Desktops to Mobile — The shift from Inside-Out to Outside-In, the corresponding explosion in data growth, and how to address this.
- Part II - Virtualizing the I/O stack — Dealing with the rise of multi-core/multi-thread, horizontally-scaled applications and the resulting impedance mismatch between compute and storage.
- Part III - IP Internetworking — Network Virtualization as the means for supporting I/O Forwarding
- Part IV - Project Autovias — Network Partitioning through virtualization.
- Conclusion

What is I/O Forwarding?

1. Bridges the gap between Compute and Storage in horizontally scaled environments

2. Introduces an intermediary tier that provides impedance matching between Compute and Storage

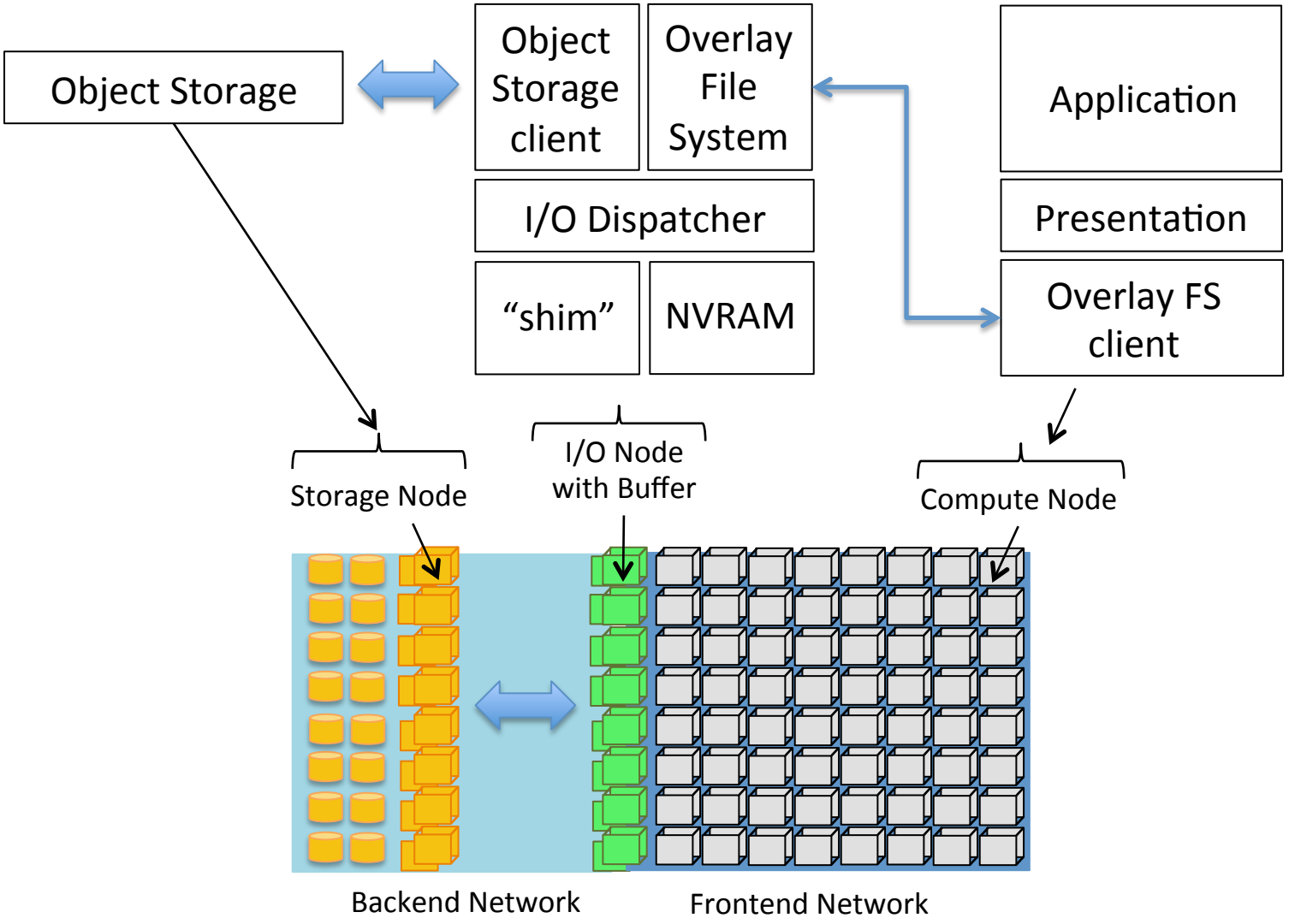
3. Nodes in this intermediary tier are dual-homed, attached to both the Compute and Storage networks, “Frontend” and “Backend” respectively.

4. Each I/O Forwarding Node presents an “Overlay File System” to nodes in the Compute tier via the frontend network while accessing Distributed Storage via the backend network.

5. Autovias focuses on managing the bandwidth available between:
    a. Compute nodes and I/O Forwarding nodes on attached to the frontend network
    b. I/O Forwarding nodes and Storage nodes attached to the backend network

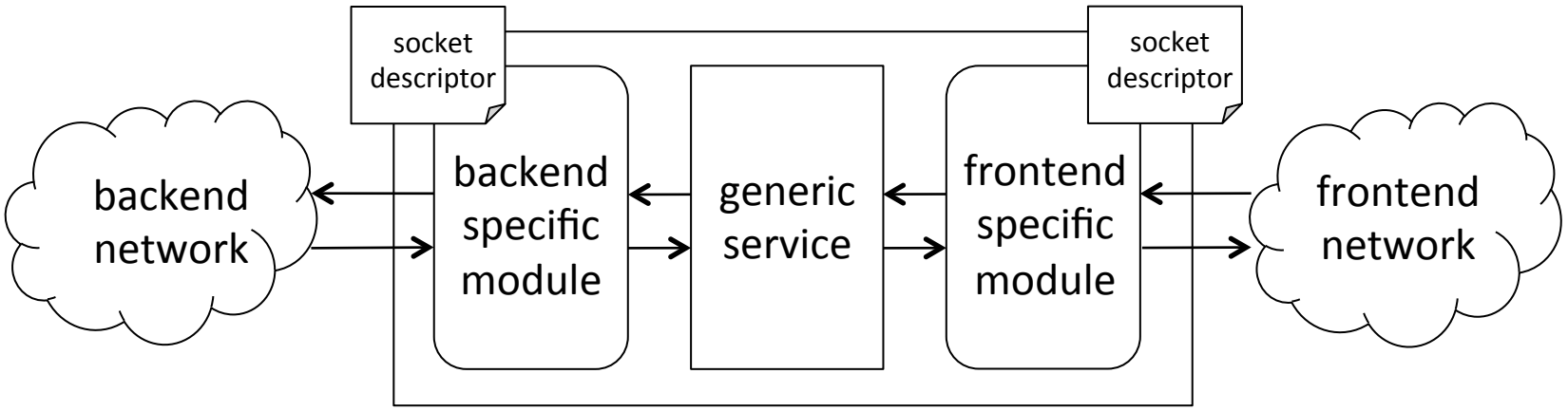
I/O Forwarding

Addressing the Impedance Mismatch between the Compute and Storage tiers



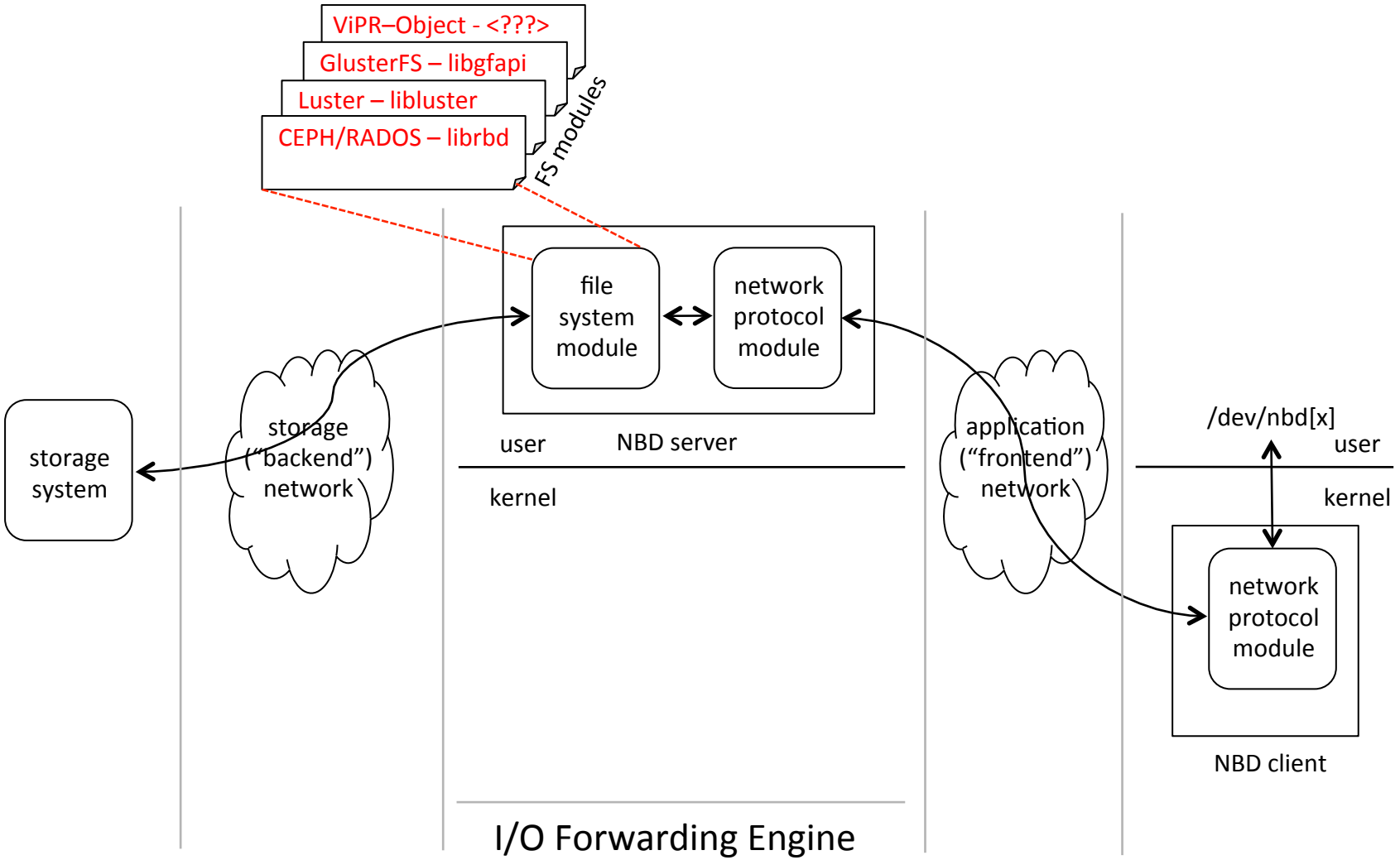
Jimenez, "The Fast-Forward I/O and Storage Stack," 2013
https://users.soe.ucsc.edu/~ivo//blog/2013/04/07/the-ff-stack/

I/O Forwarding Engine

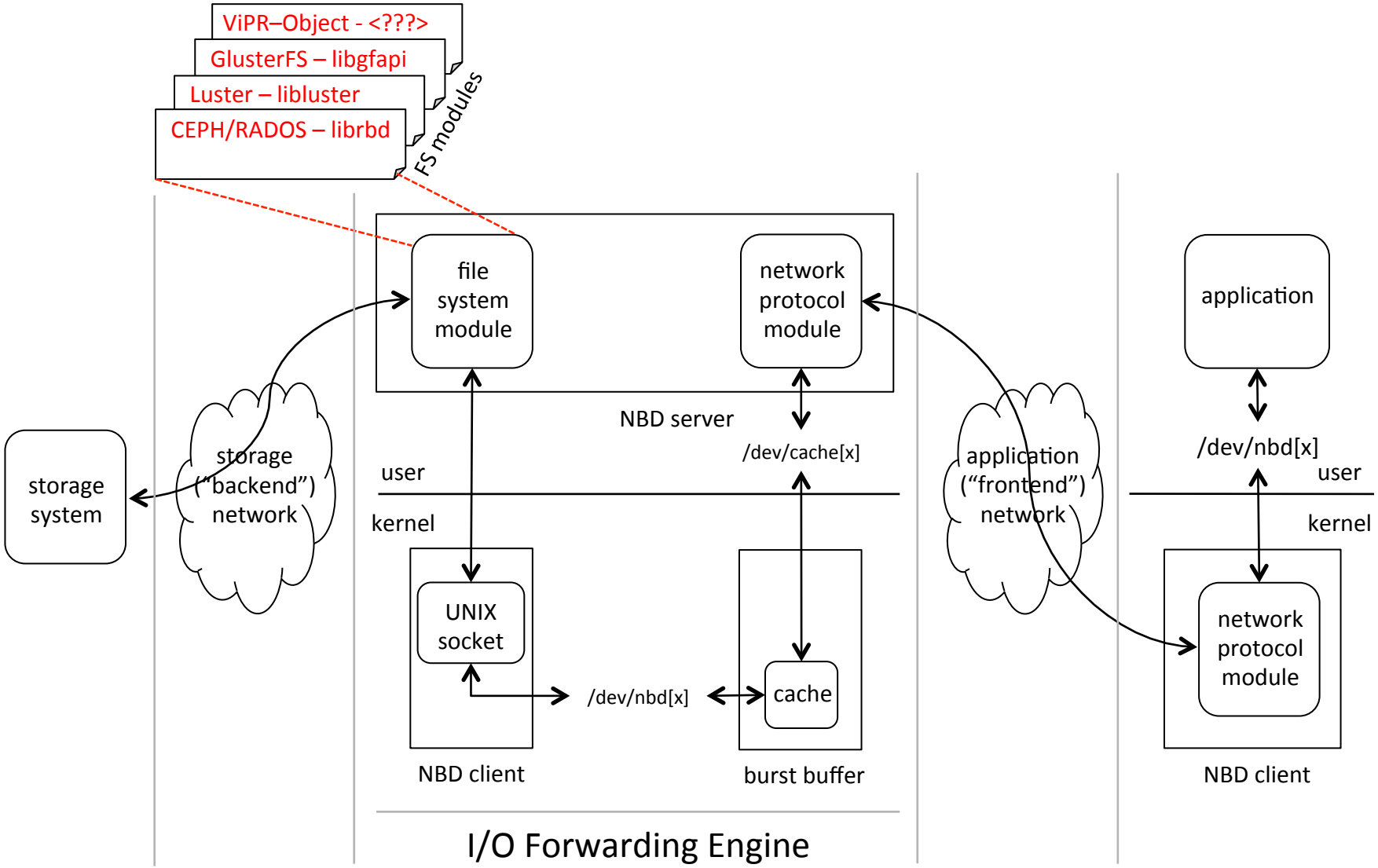


Generic Block	Network Block Device (NBD), BUSE, ploop
SCSI Block	LIO, STGT, SCST, IET, COMSTAR
File	NFS-Ganesha, DIOD, IOFSL, SAMBA/VFS, FUSE
Object	Ceph/Rados Gateway, EMC Atmos/ViPR-Object

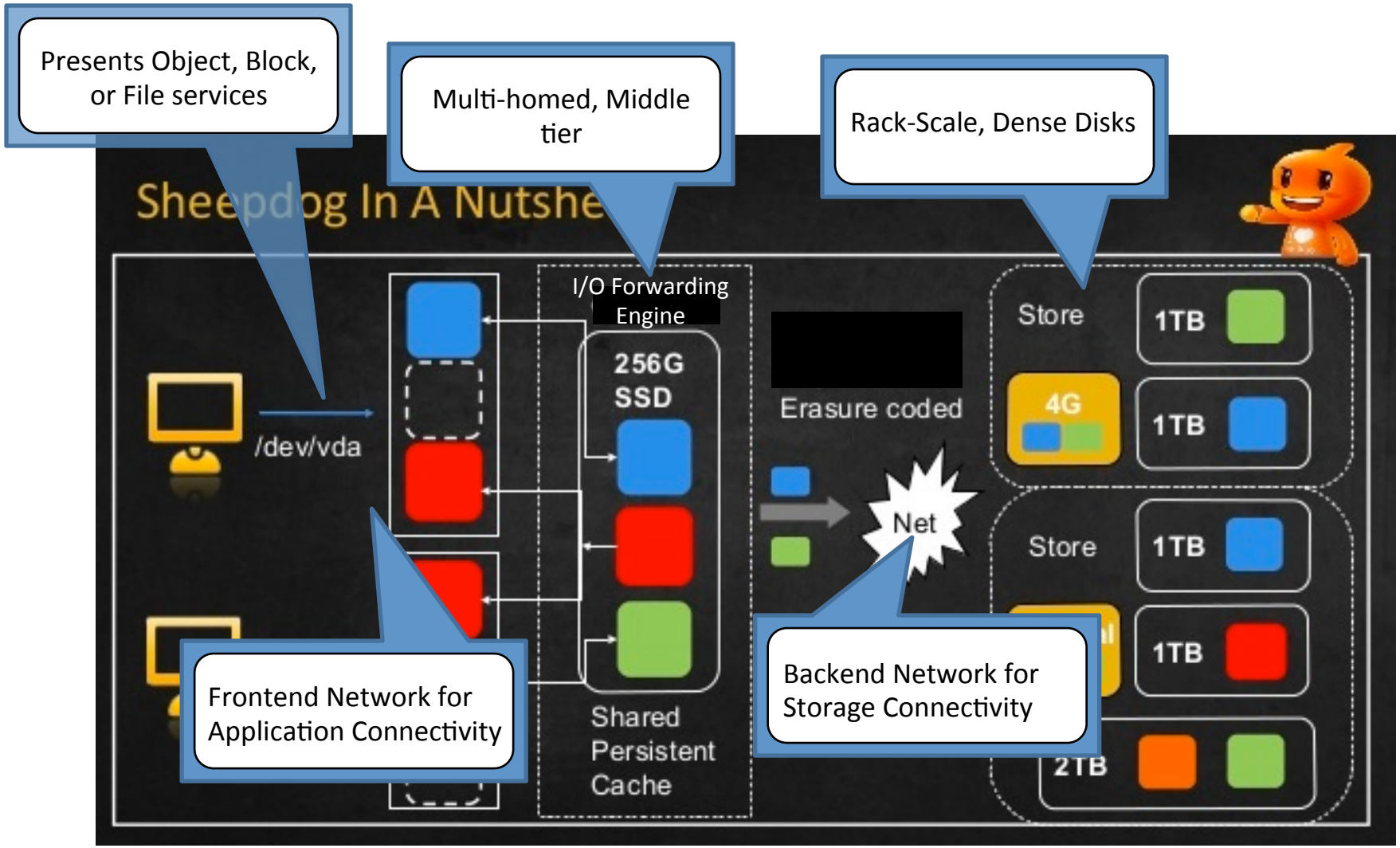
I/O Forwarding Engine (cont)



I/O Forwarding Engine w/"Burst Buffer"



I/O Forwarding - an example



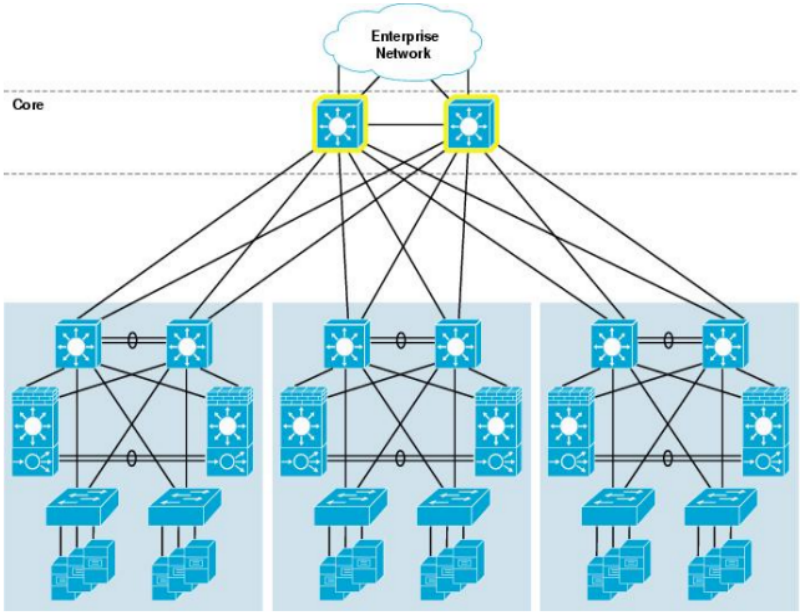
Yaun, "Sheepdog: yet another all in-one storage for openstack," 2013
http://www.slideshare.net/multics/sheepdog-yet-another-all-inone-storage-for-openstack-27402520

Agenda

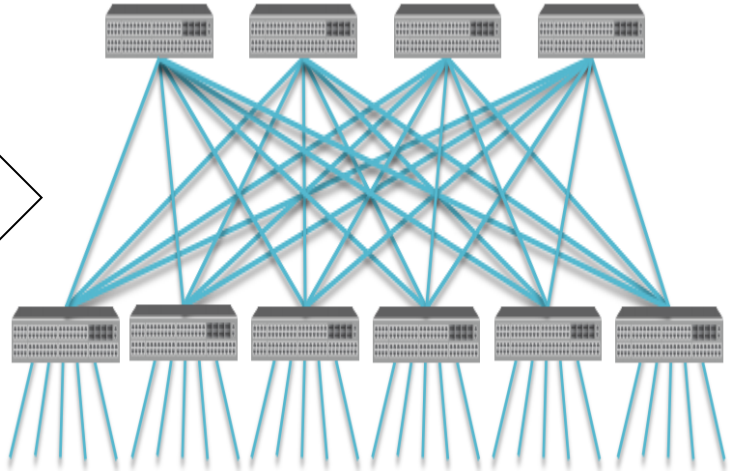
- Introduction
- Part I - From Desktops to Mobile — The shift from Inside-Out to Outside-In, the corresponding explosion in data growth, and how to address this.
- Part II - Virtualizing the I/O stack — Dealing with the rise of multi-core/multi-thread, horizontally-scaled applications and the resulting impedance mismatch between compute and storage.
- Part III - IP Internetworking — Network Virtualization as the means for supporting I/O Forwarding
- Part IV - Project Autovias — Network Partitioning through virtualization.
- Conclusion

Evolving the Data Center Network to support Scalability

1. Conventional/Enterprise Data Center Network

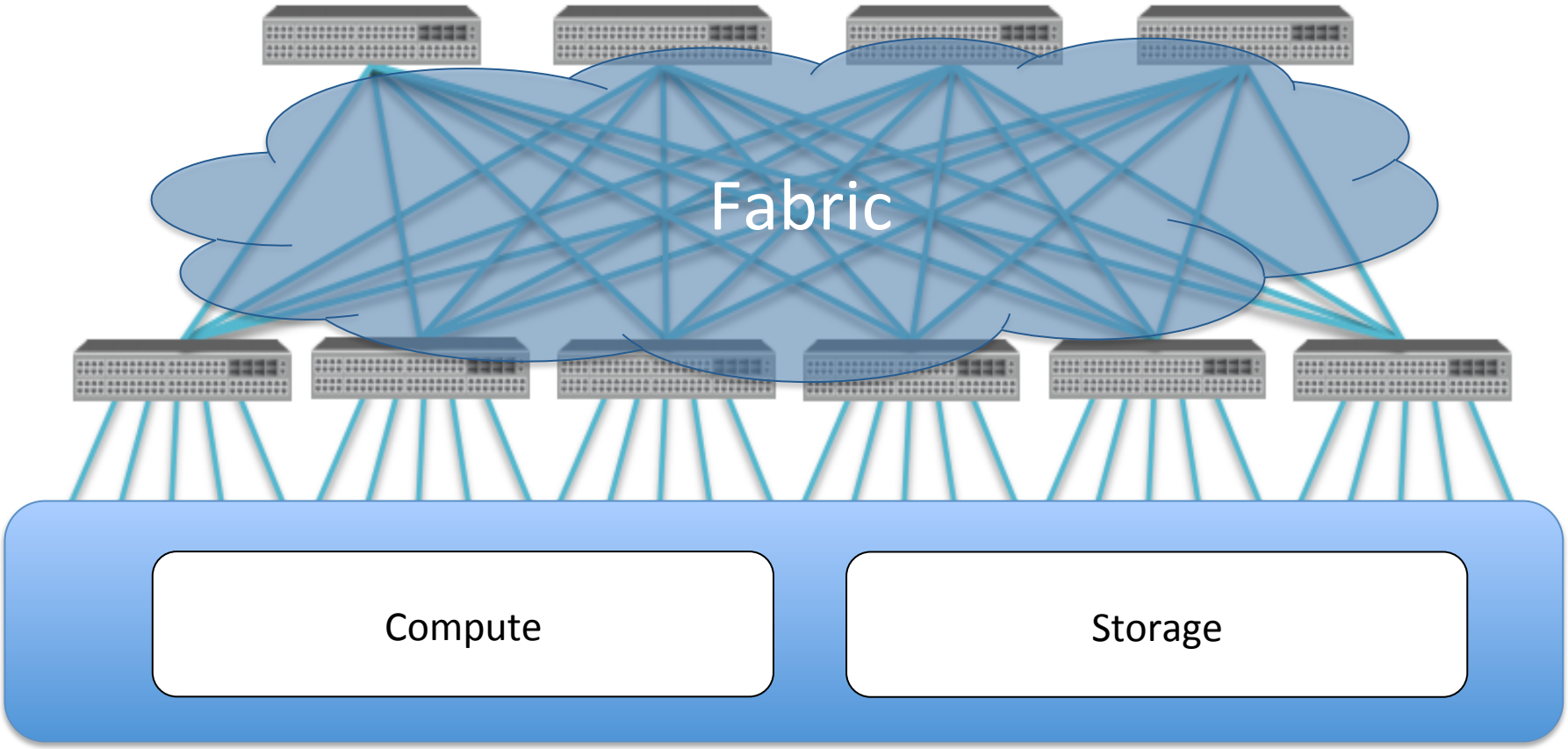


2. Scalable Data Center Network



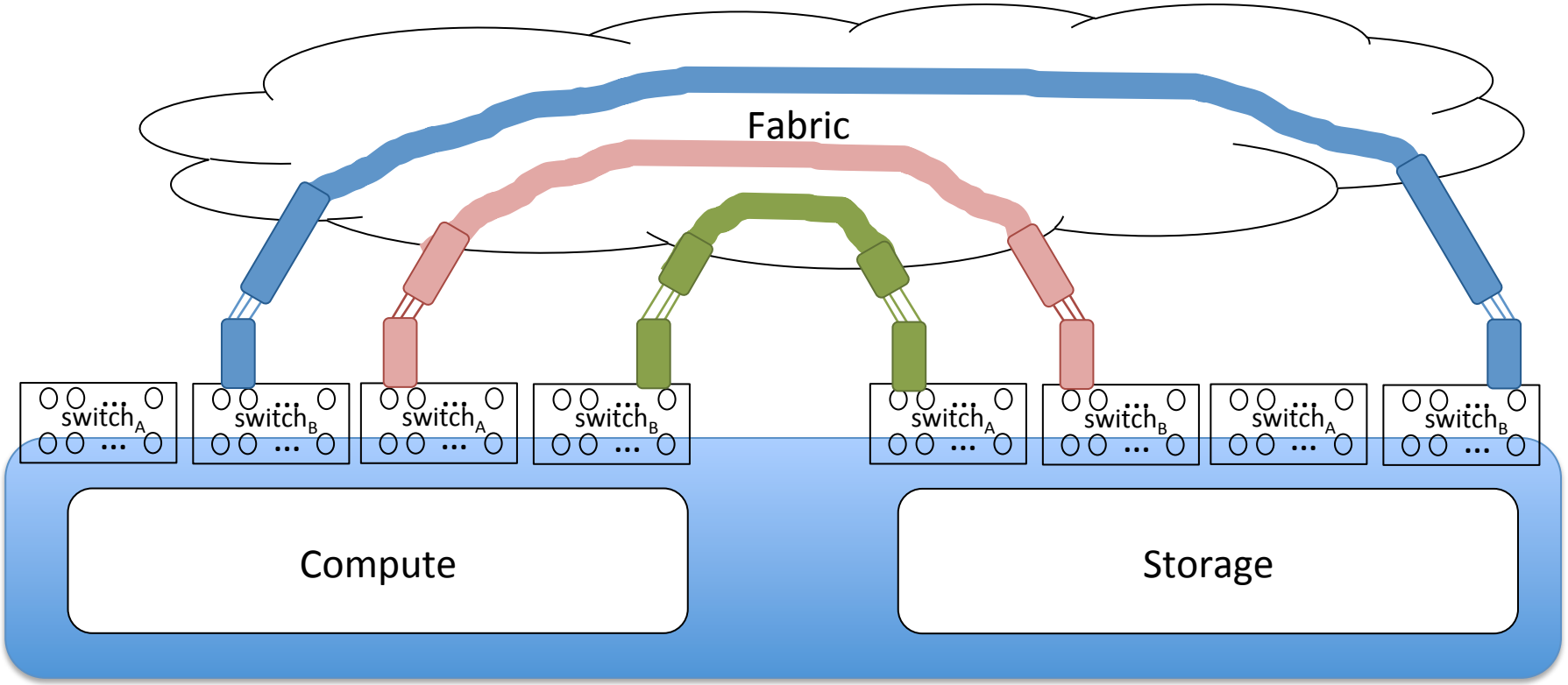
* diagrams taken Tom Edsall's "Changing Data Center" keynote presentation at the 2013 Hot Interconnects Symposium

Scalable Data Center Network

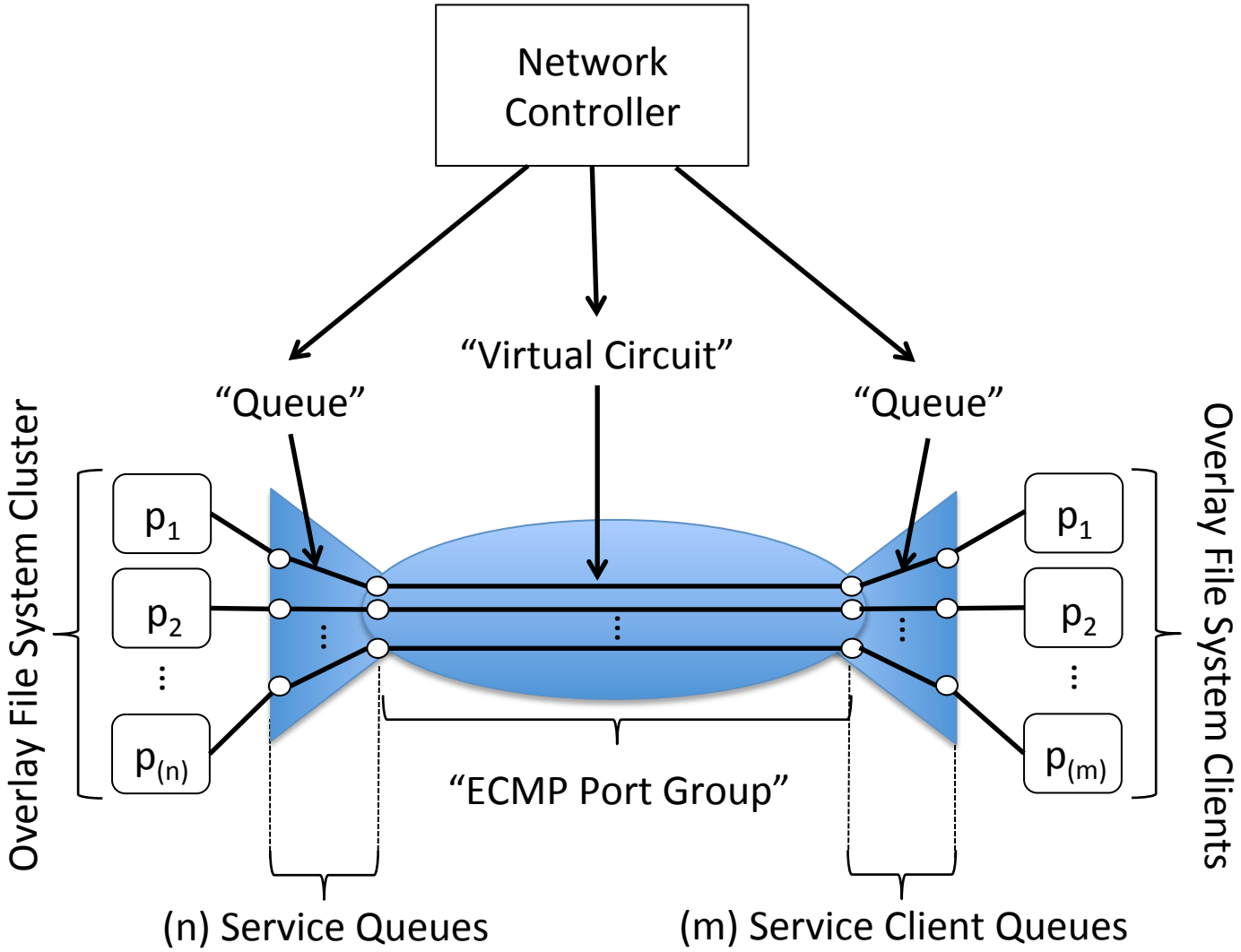


* diagram taken Tom Edsall's "Changing Data Center" keynote presentation at the 2013 Hot Interconnects Symposium

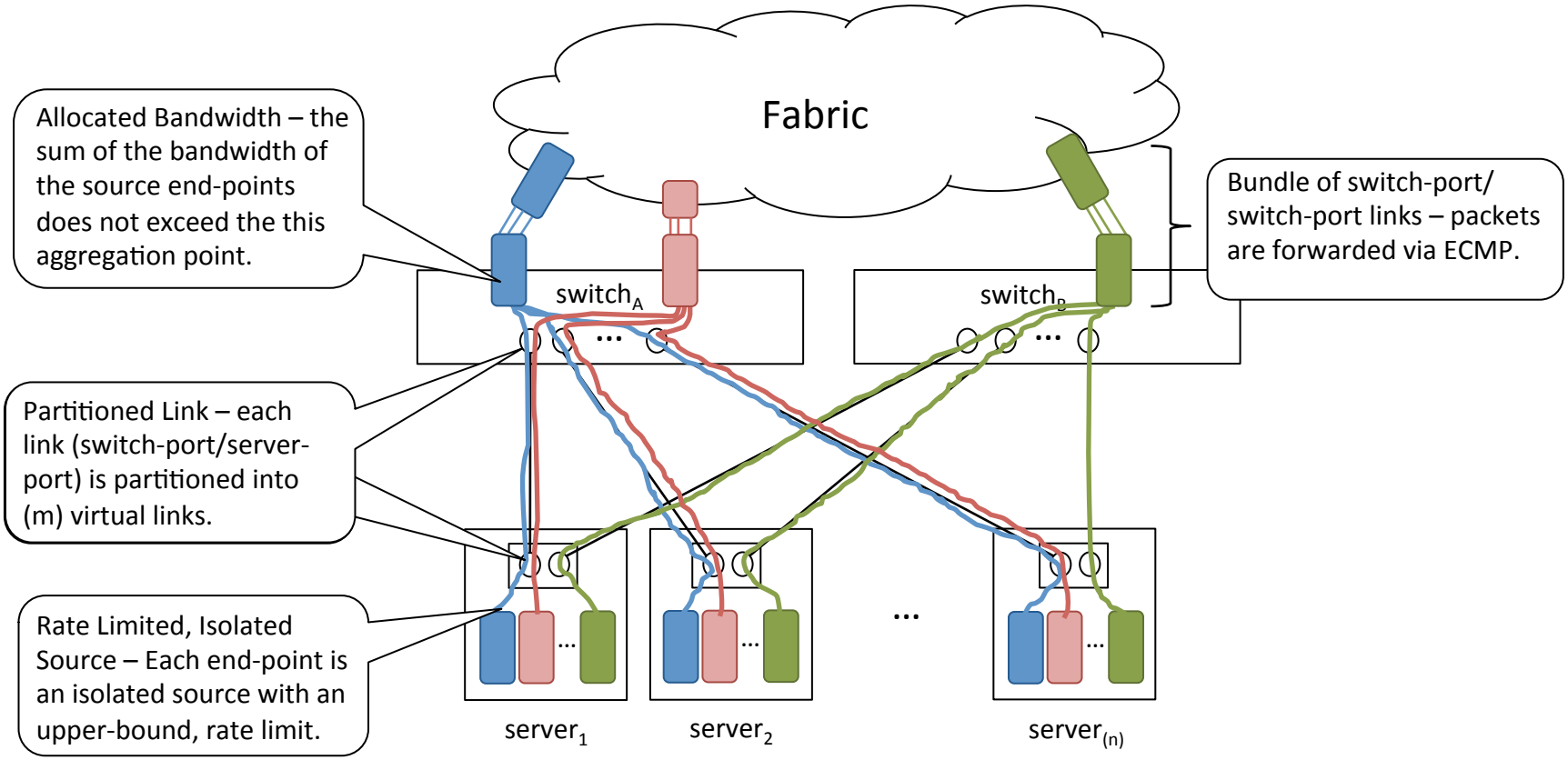
Virtual Circuits



Treating the Fabric as a Logical Switch



Bandwidth as a Schedulable Resource



Four Illustrative Deployment Scenarios

A. Google's Data Center to Data Center Connectivity [see notes 1,2]

   Google's B4 uses a combination of the ONIX controller and BGP routers to manage ECMP port groups. Each ECMP port group is a set of virtual circuits that provide connectivity between two of BGP routing domains. . In this context, the ONIX controller is a "Route Server" that speaks BGP and peers with the routers in each of several AS routing domains. For a given pair of routing domains there is an ECMP port group managed by the ONIX controller.

B. Microsoft's BGP-Only/SDN-Controlled Data Center Network [see notes 3,4]

   Microsoft has developed a BGP-only data center network that combines an ONIX-style controller and BGP routers. Similar to Google's B4, Microsoft's ONIX-style controller speaks BGP and peers with routers in the data center network for the purpose of managing ECMP port groups.

C. Open Network Lab (ONL)'s BGP-Only/SDN Controlled Internet Exchange Point (IXP) [see notes 5,6]

   Yet a third illustration of this technique can be seen in the Open Networking Tree-House project. Like the Google and Microsoft approaches above, this is based on ONOS, an ONIX-style controller and BGP routers. ONUS speaks BGP and peers with routers in an Internet Exchange Point (IXP).

D. BGP-Only/SDN Controlled Network Architecture [see note 7]

   Finally, Ivan Pepelnjak recently posted an article to his blog about building a BGP-only data center using the combination of a network controller that acts as a BGP route server and a set of BGP routers. As you can imagine, the controller Ivan imagines is one that speaks BGP and peers with the routers in a manner similar to the other three examples above.

Candidate Network Virtualization Controllers

1. Google's fork of the ONIX Controller
   - [Proprietary/Service-Only]
   - [notes 2,3]

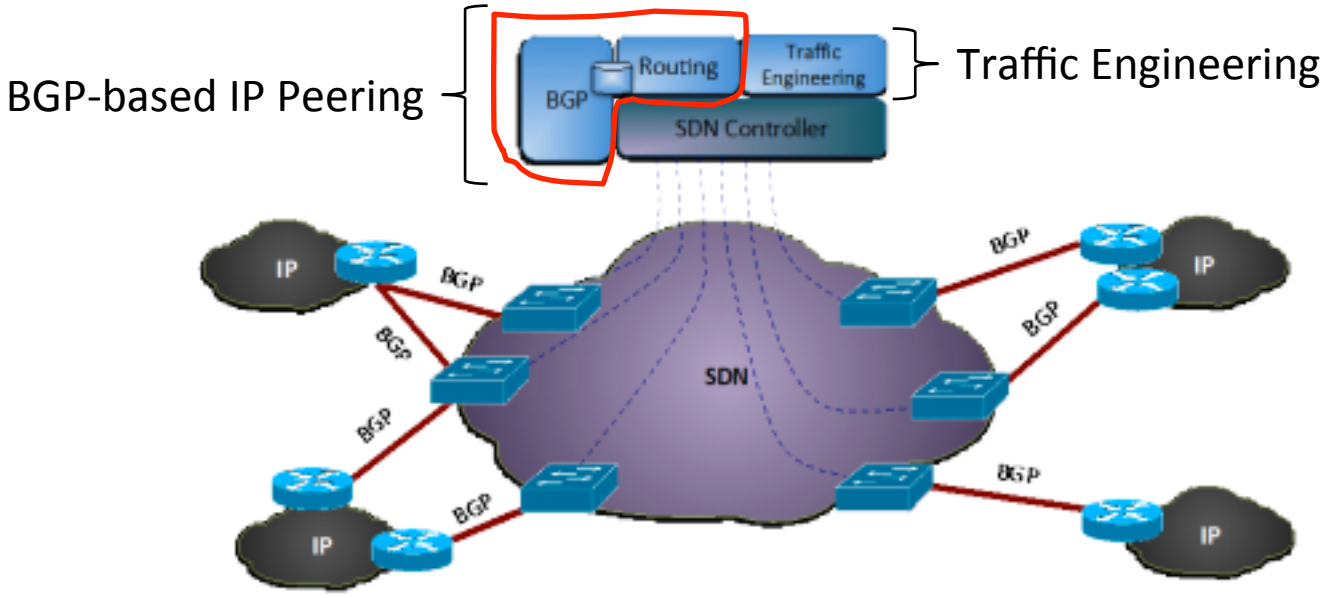
2. Vmware's NSX Controller, also a fork of the ONIX Controller
   - [Proprietary/Commercial Product]
   - [notes 2,4]

3. Microsoft's BGP-Only SDN
   - [Proprietary/Service-Only]
   - [notes 2,5,7,8]

4. Open Network Lab's Open Network Operating System (ONOS)
   - [note 9, 10, 11]

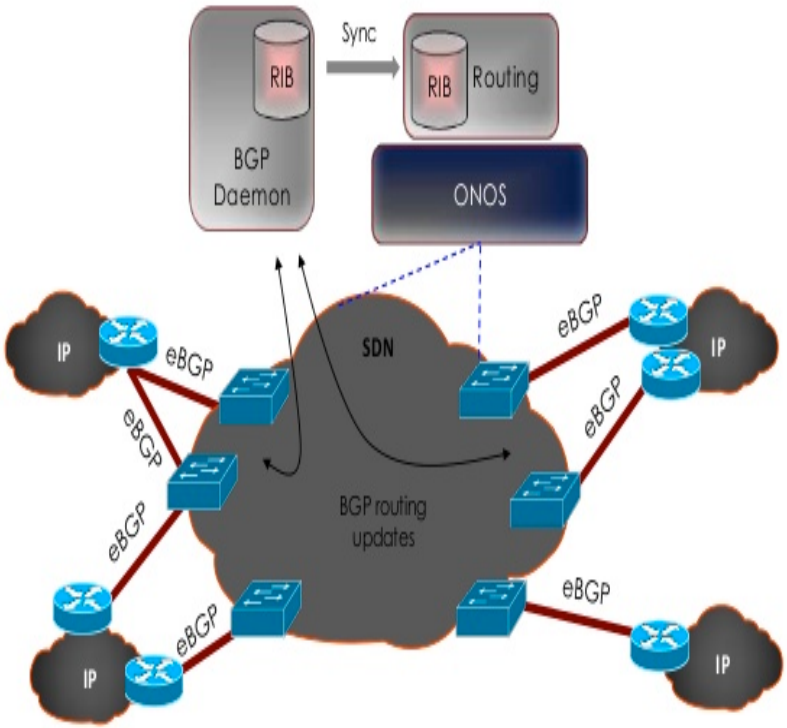
Open Network Operating System (ONOS)

1. Scale-out Architecture
   - allows servers to be added or deleted to match control traffic demand.
2. High Availability
   - combines Cassandra (distributed hash table-based storage) and Zookeeper (distributed registry) to store network state and configuration.
3. Network Graph Abstraction
   - northbound API leveraging the Titan distributed graph database to enable the Topology Viewer, Path Computation, and SDN-IP Peering applications.

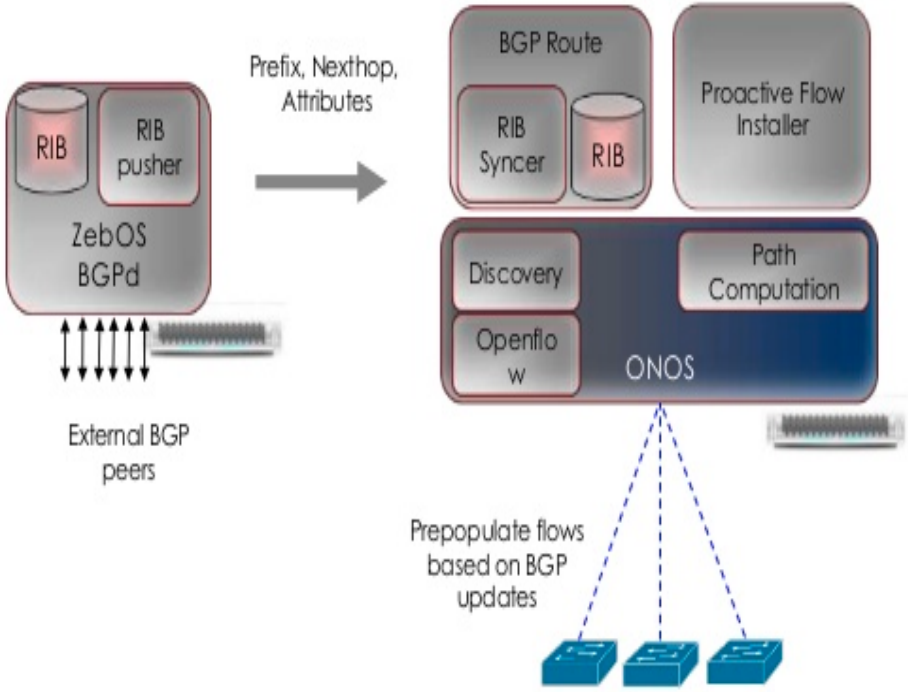


SDN-IP – a Peering Application using ONOS

1. ONOS as a BGP Route Server



2. ONOS as a BGP Route Server



Use BGP Peering and Route Server to interact with the network, no OpenFlow!

Agenda

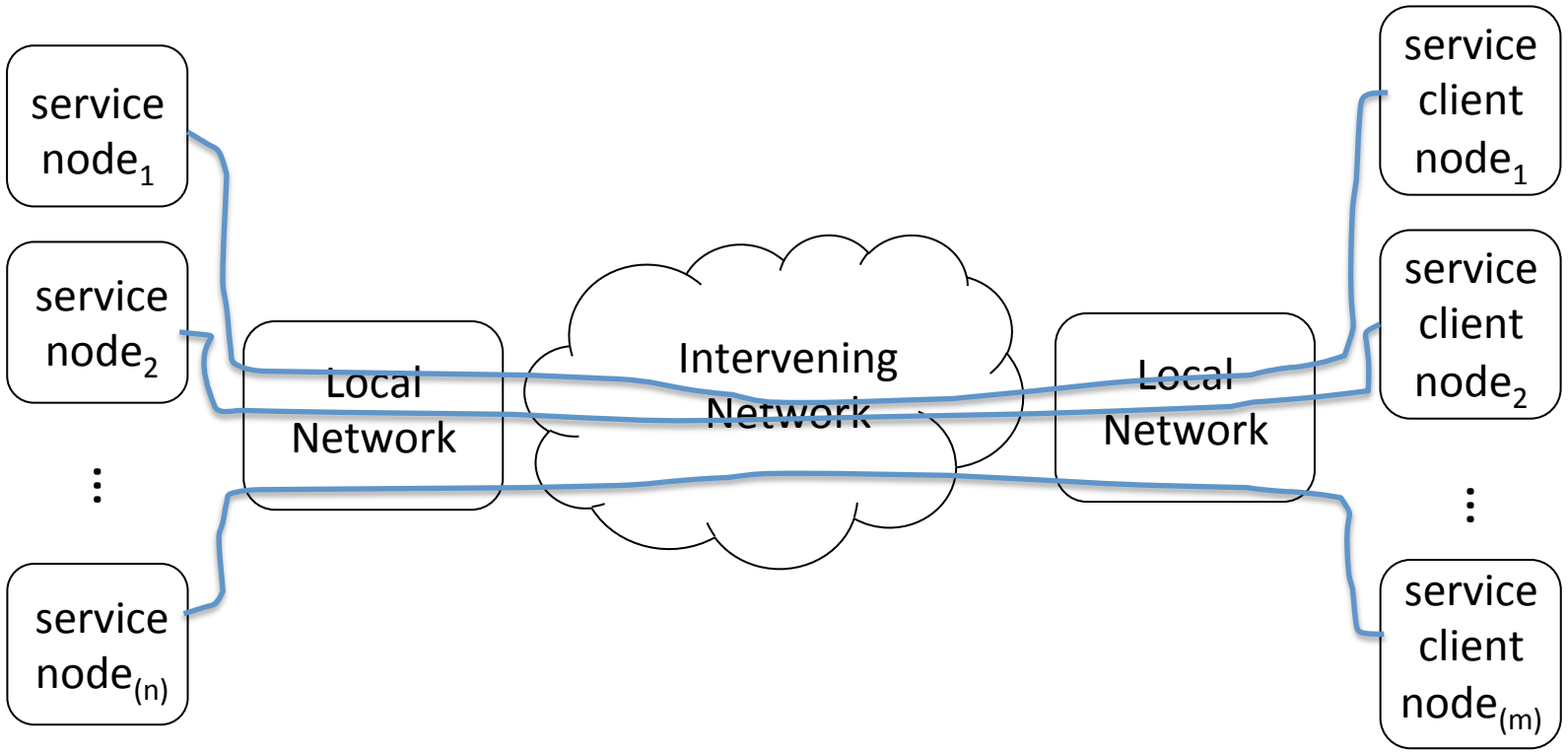
- Introduction
- Part I - From Desktops to Mobile — The shift from Inside-Out to Outside-In, the corresponding explosion in data growth, and how to address this.
- Part II - Virtualizing the I/O stack — Dealing with the rise of multi-core/multi-thread, horizontally-scaled applications and the resulting impedance mismatch between compute and storage.
- Part III - IP Internetworking — Network Virtualization as the means for supporting I/O Forwarding
- Part IV - Project Autovias — Network Partitioning through virtualization.
- Conclusion

Autovias – a network partitioning scheme for I/O forwarding

Autovias Background

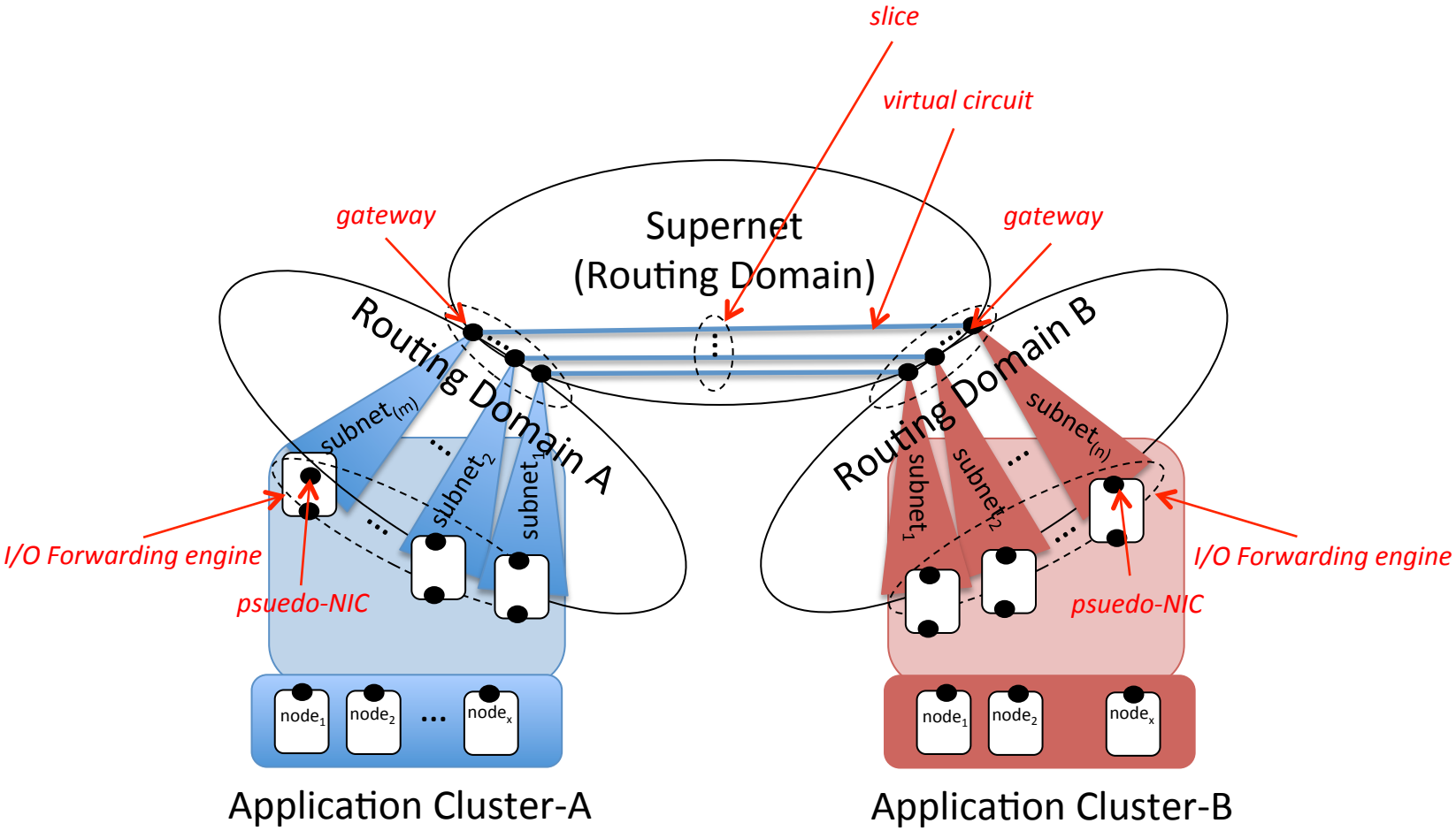
1. Addresses issues that arise from Storage and Replication traffic over shared networks.
2. Autovias is not a product. It is technology targeted enhancing solutions such as
   - Lustre, Atmos/ViPR-Object, etc
   - VPLEX, RecoverPoint, DataDomain
3. Allows a user/service manager to schedule network bandwidth for their service.
4. Concept was conceived in 2012 and derives from Google's B4 Data-Center-to-Data-Center solution (as presented by Urs Hölzle at the Open Networking Summit 2012)
5. Employs a Network Virtualization Controller such as Vmware's NSX Controller
6. First, demonstrated and discussed publically at EMC World in May 2013
7. Subsequent demonstrations at Vmworld, August 2013 and Vmworld Europe, October 2013

Autovias Problem: (m) service client to (n) service node connectivity

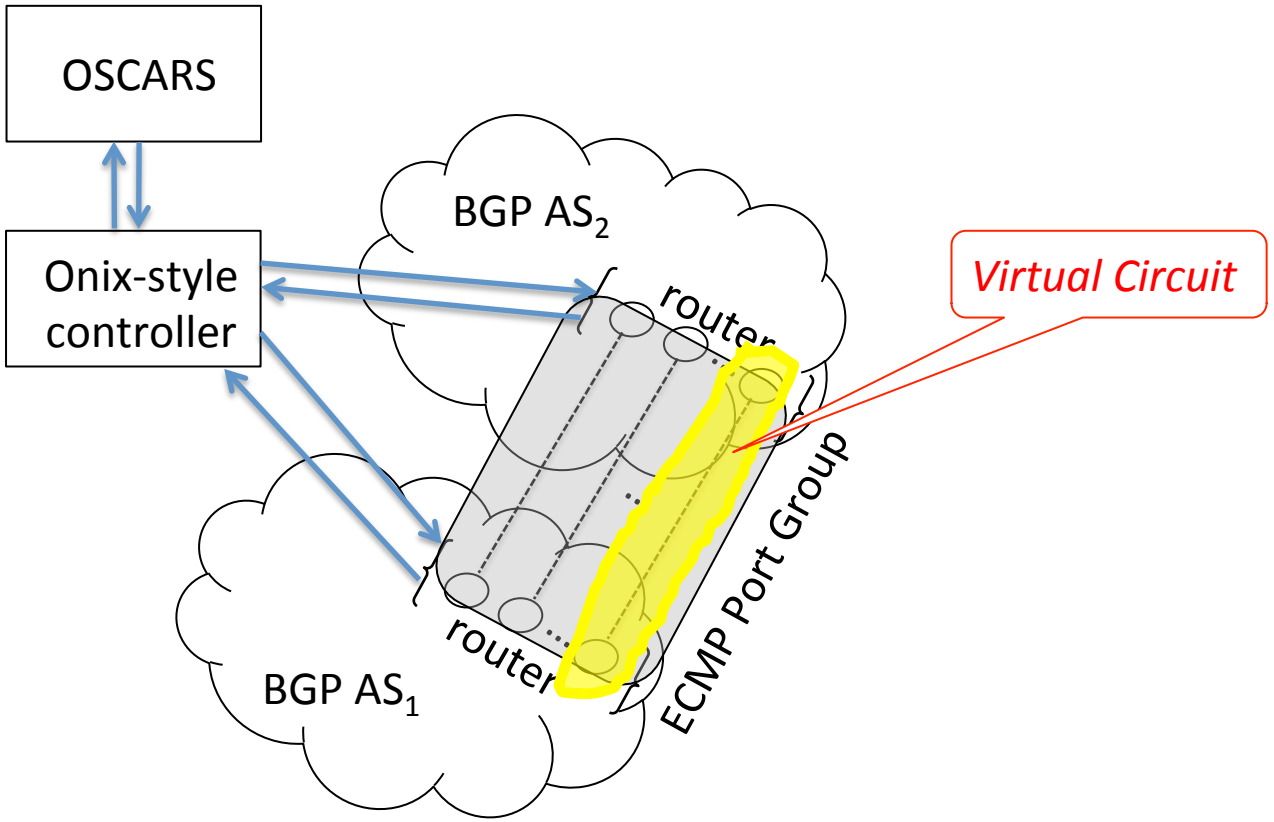


Can we model the ideal network as a “Logical Crossbar Switch?”

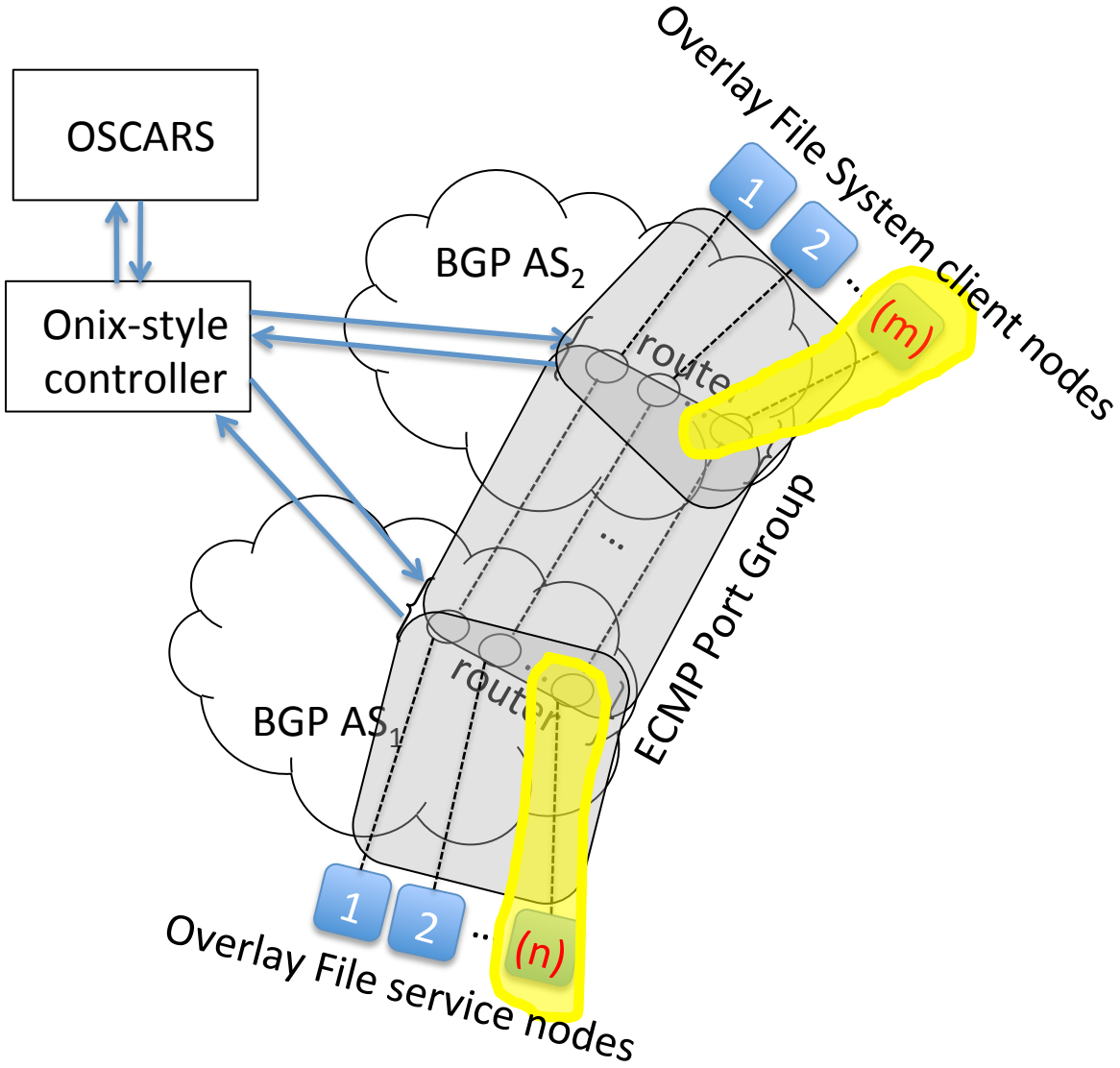
Bringing it all together



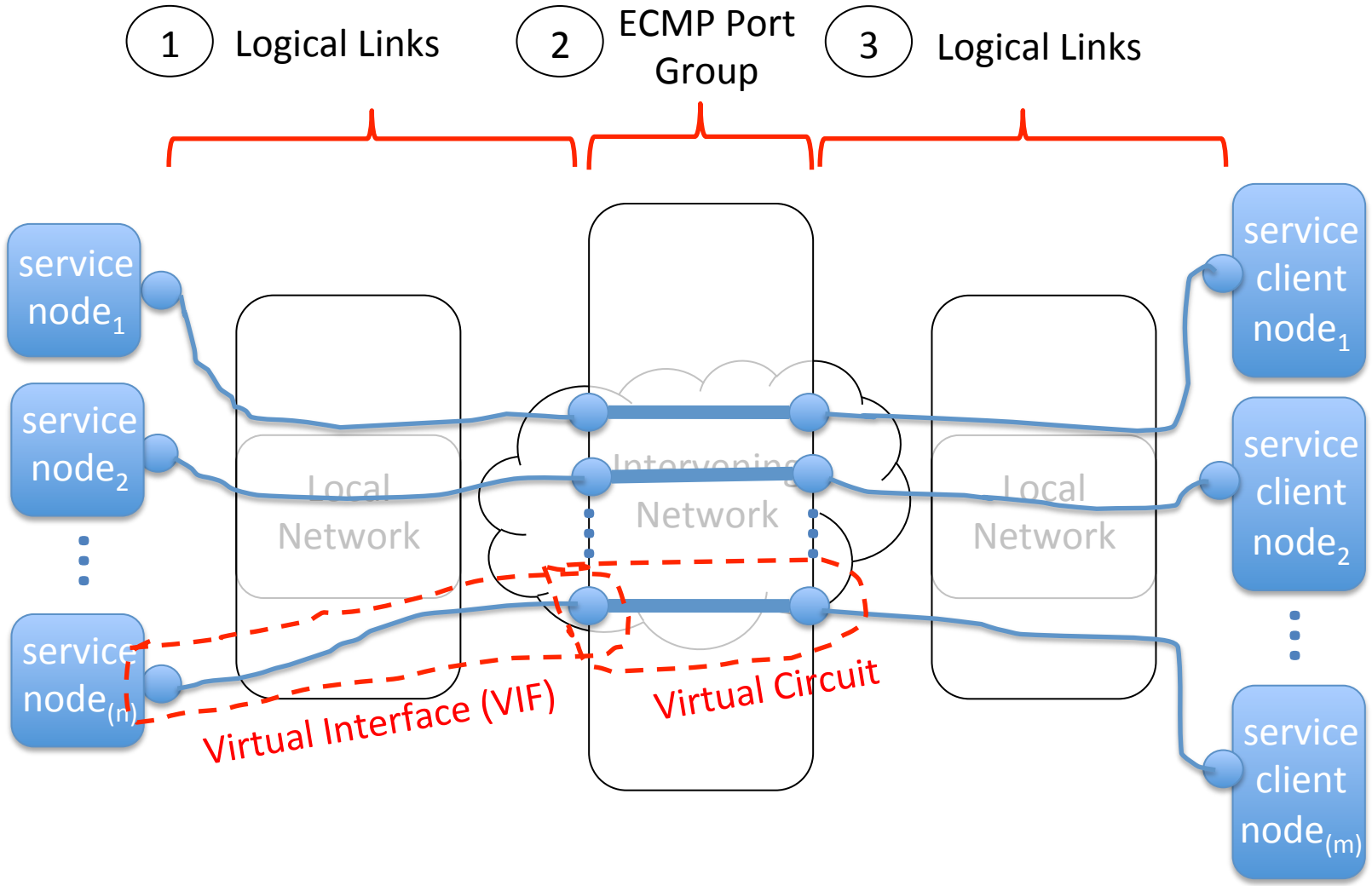
Step I - Use OSCARS in conjunction with an ONIX-style, SDN Controller



Step II – Use a Server-Resident Pseudo Bridge with an Onix-style, SDN Controller (cont)



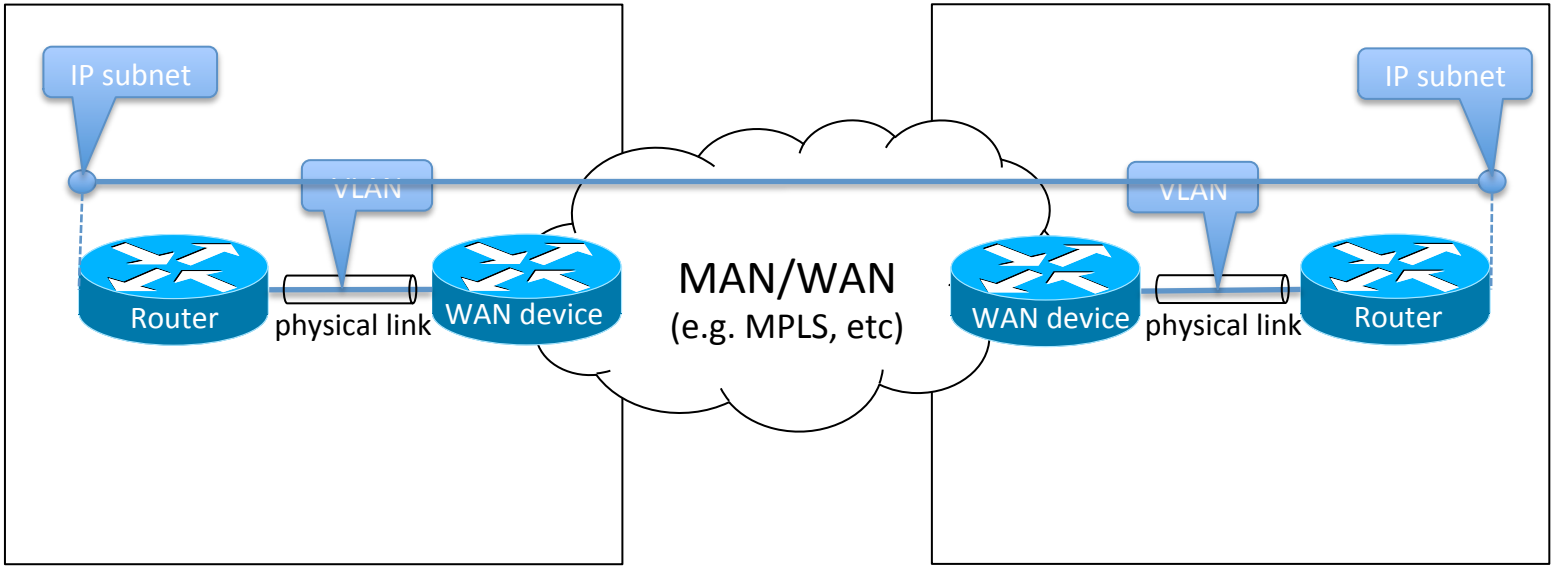
Autovias Solution: Logical Switch

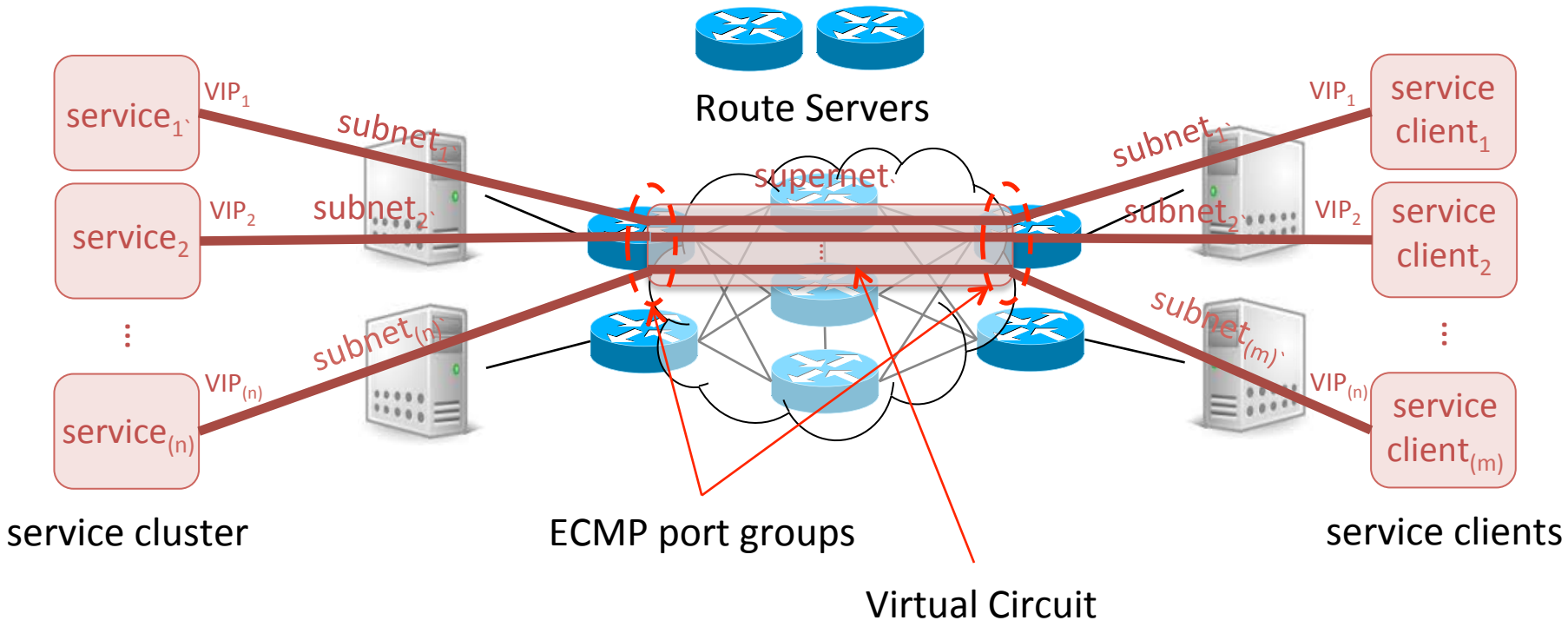


Requires two classes of "IP Routes:" (i) Virtual Circuit and (ii) Virtual Interface (VIF).

Autovias Virtual Circuit concept using SDN-IP Peering

1. A Virtual Circuit is a flow between two IP subnets as depicted below.
2. These subnets are expressed in CIDR summary notation with the same “block” size.
3. Both subnets are prefixes from the same level in a supernet (e.g. /24s in a /20 network.
4. There *MUST* be the same number of “OSCARS” virtual circuits (ie WAN) as the block size.
5. ECMP is used to spread traffic across the these paths.

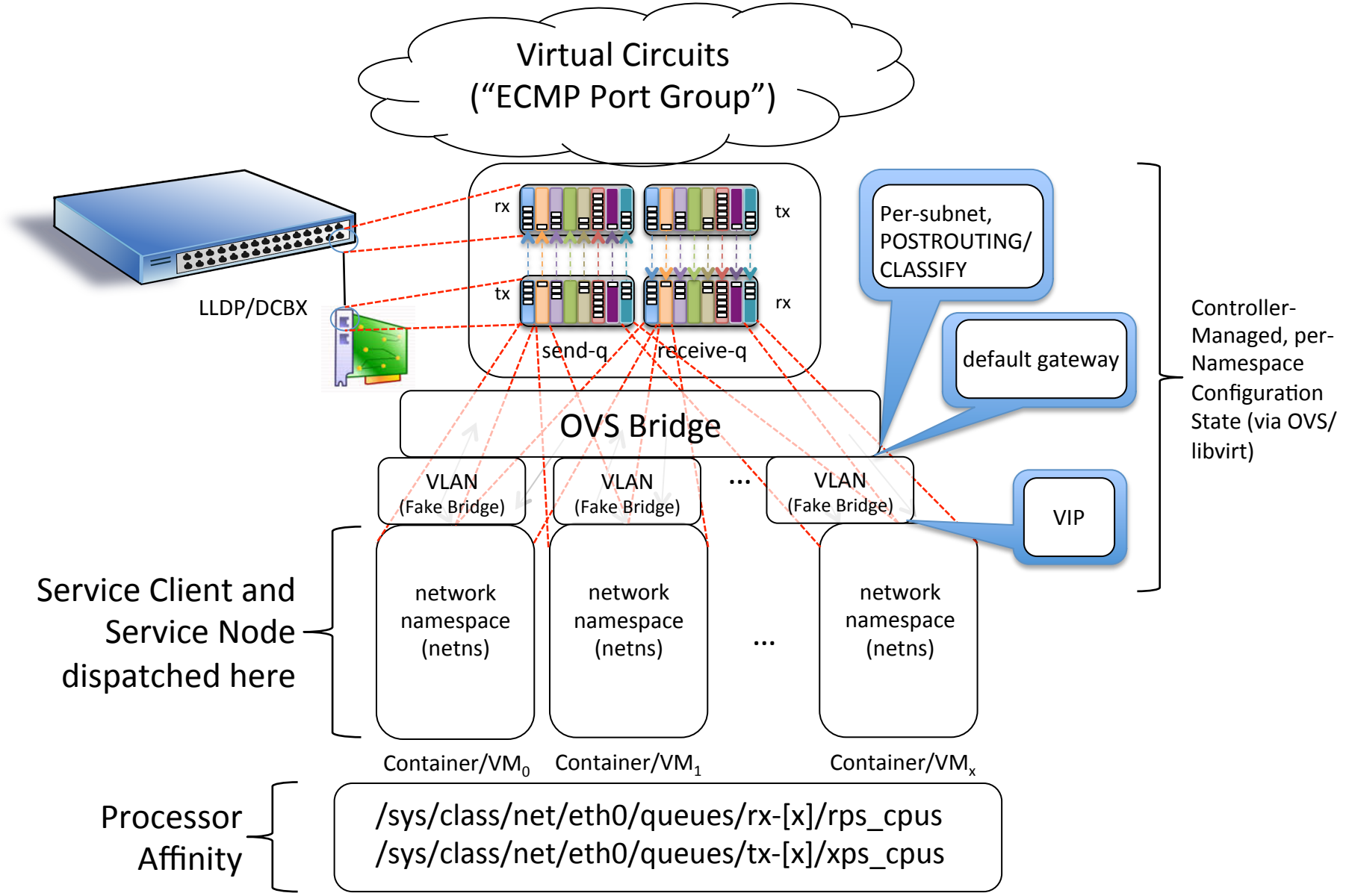




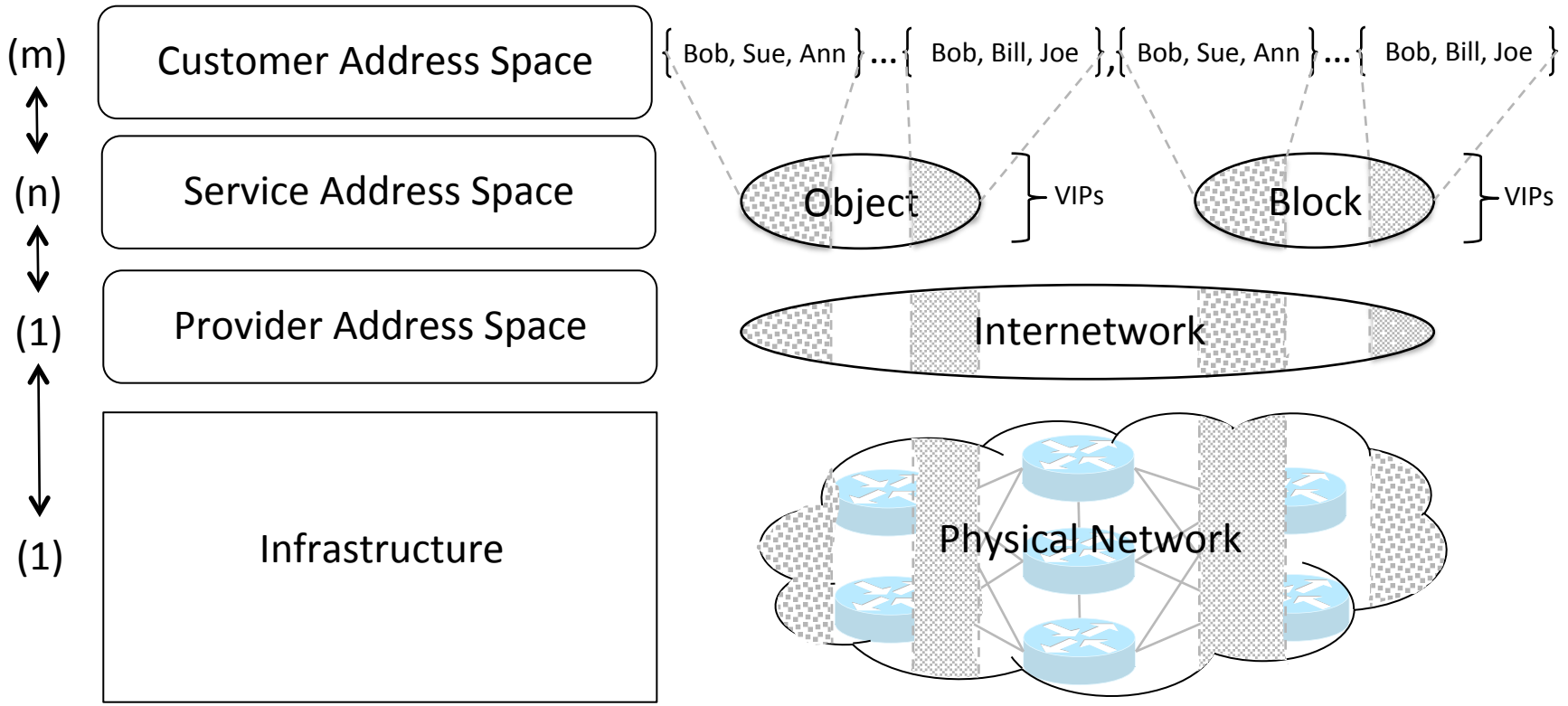
Bernat, "High availability with ExaBGP," 2013
http://vincent.bernat.im/en/blog/2013-exabgp-highavailability.html

Using BGP to serve High-Availability DNS," 2012
https://www.iweb-hosting.co.uk/blog/using-bgp-to-serve-high-availability-dns.html

Autovias Virtual Interface Concept



When multiple services share a network



Closing Thoughts