Perspective of Virtual Switching Fabric

jiezheng@vmware.com

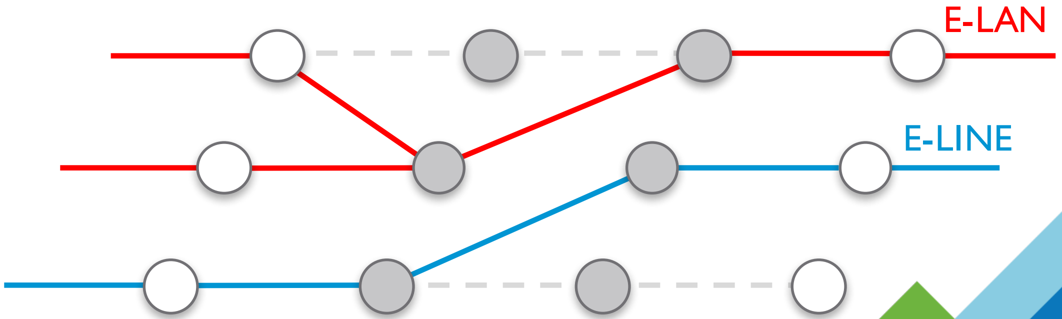
Agenda

- ❑ **Overview of Fabric Structure**
- ❑ Emulation & Demonstration
- ❑ Performance Evaluation
- ❑ Use cases Imagination

What is the Fabric?

A L2 virtual networking using smart routing

- node naming: spine vSwitch node vs leaf vSwitch node
- pseudo wire service: Ethernet-Line(E-Line)
- pseudo lan service: Ethernet-LAN(E-LAN)
  - *mac based forwarding*
  - *emulate LAN at the core(built-in multicast tree, replication on-demand)*
- path optimization
  - *adjacency next-hop count as weight*
  - *link workload as weight*
  - *for E-LAN, minimum spanning tree*
  - *For E-LINE, shortest path*

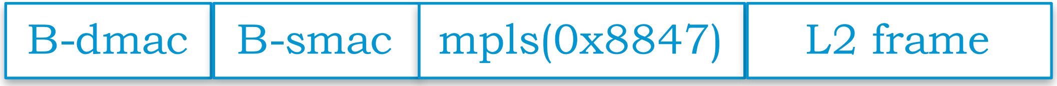


vmware

What is the Fabric? contd

transport layer consideration

- Ethernet over MPLS ? RFC448
  - *path selection and tenant identification*
  - *encapsulation overhead: 12+2+4=18 bytes*
- mitigate TOR variables:
  - *PF and VF mac addresses.*
- hardware independency
  - *NICs supported by DPDK*
  - *high volume switch*



B-dmac	B-smac	mpls(0x8847)	L2 frame

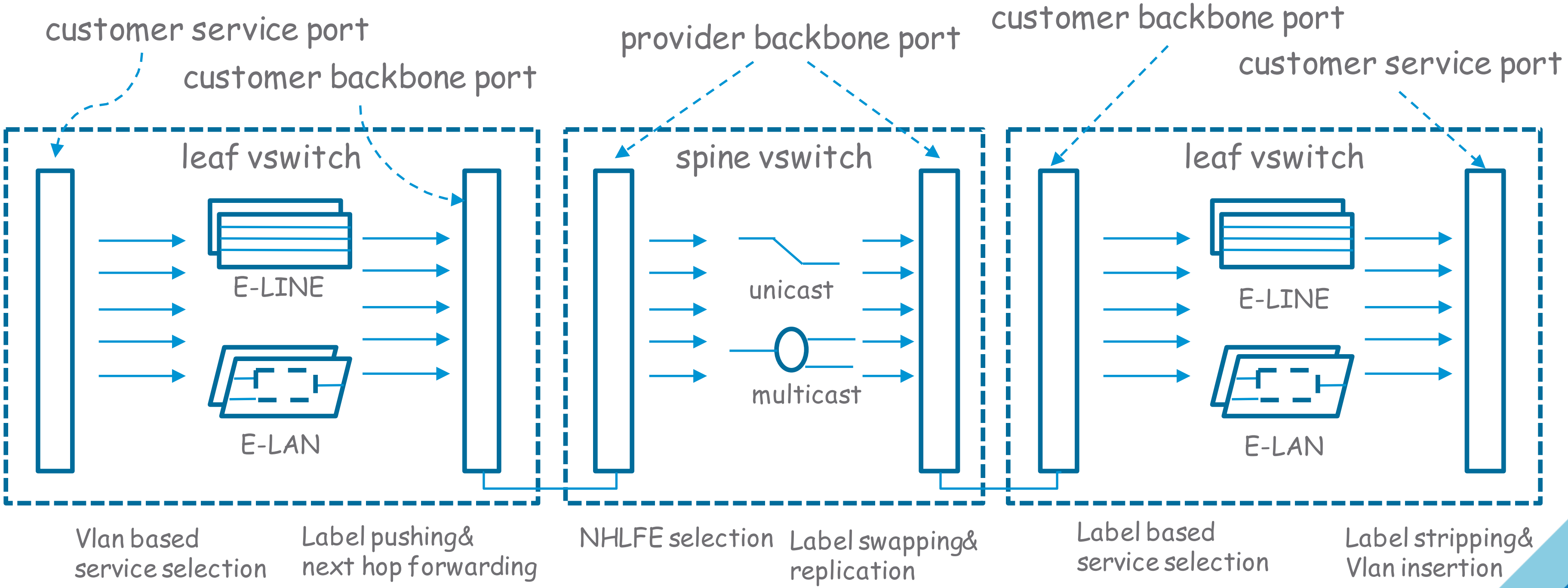
Why Fabric?

x86 platform networking capability

- data path essence
  - *hardware IO capability(PCIe Gen3 x8GT/s TL bandwidth utilization :66%) from infrastructure network into host(hypervisor) memory vpp benchmark: 480Gbps L3 forwarding, does not scale any more due to nic/PCI bus limitation*
  - *memory bandwidth is bottleneck of virtual network from vm to vm(host/hypervisor) experiment: 60Gbps intra-numa-node ,two times memory copy, does not scale well memory movement consumes too much cpu time and memory bandwidth*
- let dedicated servers as fabric do virtual networking!
- more agility and hardware dependency than specific hardware solution.
  - *as the intrinsic requirement of NFV*

Fabric constitution

virtual switch internals

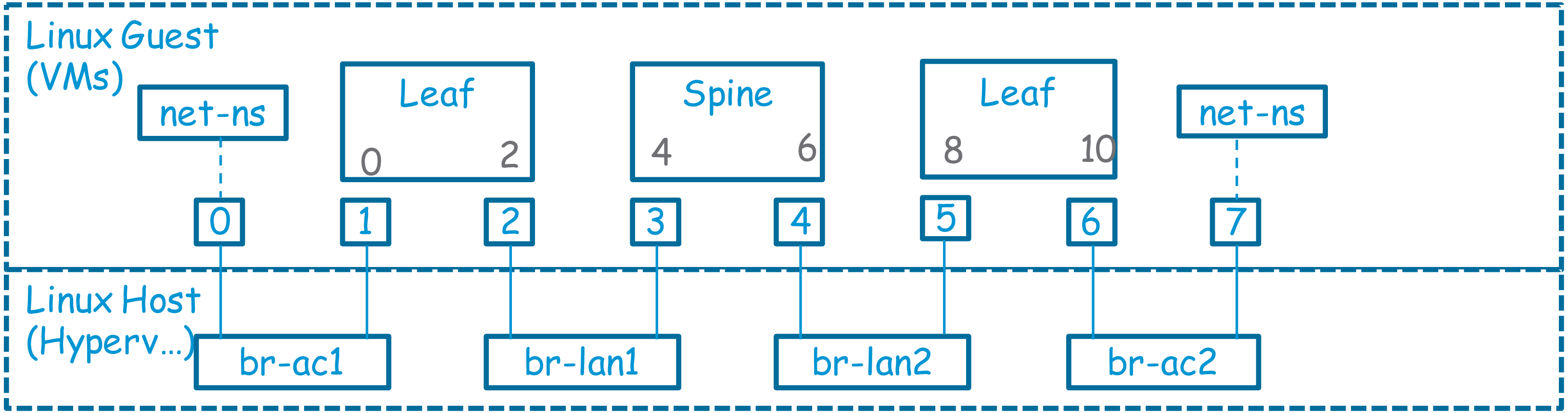


Agenda

- ❑ Overview of Fabric Structure
- ❑ **Emulation & Demonstration**
- ❑ Performance Evaluation
- ❑ Use cases Imagination

Emulated environment pre-setup

- Qemu KVM emulated guest, both host and guest OS are CentOS 7
- 8 x e1000 devices with VLAN stripping and insertion offloading capability
- 4 x ovs bridges for 4 LANs emulation(2 attachment circuit lans and 2 common lans)
- Port 0 and port 7 are for customers and port 1-6 are for fabric switches
- For fabric ports, once taken over by e3datapath, re-index them [0,2,4,6,8,10]
- For customer ports, use Linux namespace and vlan sub-interfaces to segregate themselves



E-line service

csp port 0	cbp port 2	pbp port 4	pbp port 6	cbp port 8	csp port 10
create two e-line services: 0 and 1					
Vlan1000 ----> e-line 0					
	E-line 0 next hop(to port 4) via port2 with label:1				
		Port 4 with label 1,knows next hop(to port 8) via port 6 with label 100			
				Port 8 with label 100 goes to e-line1	
					e-line1--->vlan 2000
					vlan 2000--->e-line1
			E-lan1 next hop(to port 6) via port 8 with label:10000		
		Port 6 with 10000,it knows next hop(to port 2) with label 10.			

E-lan service multicast forwarding

<table>
<tr><th>csp port 0</th><th>cbp port 2</th><th>pbp port 4</th><th>pbp port 6</th><th>cbp port 8</th><th>csp port 10</th></tr>
<tr><td colspan="6">create two e-lan services: 0 and 1 , and multicast next hop list:0</td></tr>
<tr><td>Vlan3000 ----> e-lan 0</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td colspan="2">E-lan 0,find no fwd entry, multicast next hop(to port 4) via port2 with label:2</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="2">Port 4 with label 2, goes to multicast list0,perform RPF check and send replication (to port 8) via port 6 with label:101</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td>Port 8 with label 101 goes to e-lan1</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>e-lan1--->vlan 4000</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>vlan 4000--->e-lan1</td></tr>
<tr><td></td><td></td><td></td><td colspan="2">E-lan1 still finds no fwd entry, use multicast nexthop(to port 6)via port 8 with label:10001</td><td></td></tr>
<tr><td></td><td></td><td colspan="2">Port 6 with 10001,does multicast forwarding, finally goes to port 2 via port 4 with label:11</td><td></td><td></td></tr>
</table>

E-lan service unicast forwarding

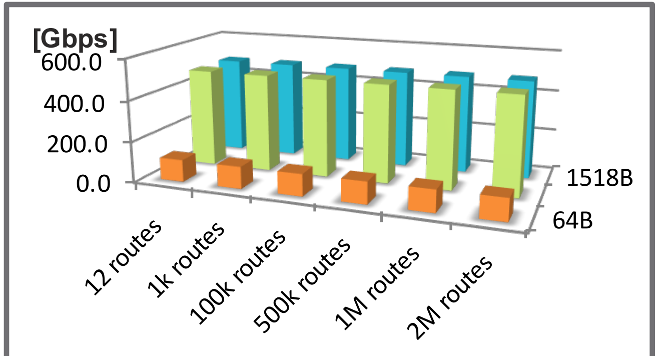
- At leaf virtual switch, a fwd entry is found with deterministic <label, nhlfe>
- At spine virtual switch, single next hop is bound to input label entry, no multicast list is searched

Agenda

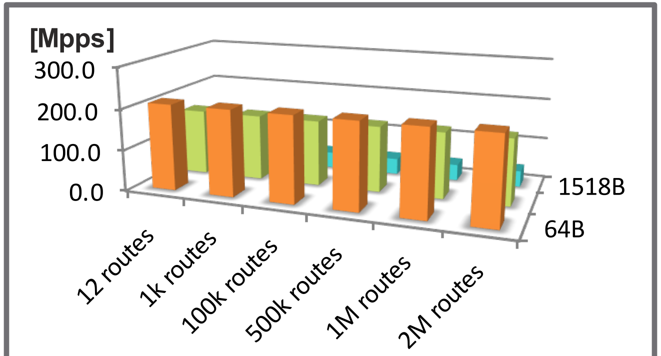
- ❑ Overview of Fabric Structure
- ❑ Emulation & Demonstration
- ❑ **Performance Evaluation**
- ❑ Use cases Imagination

VPP Performance at Scale

IPv6, 24 of 72 cores

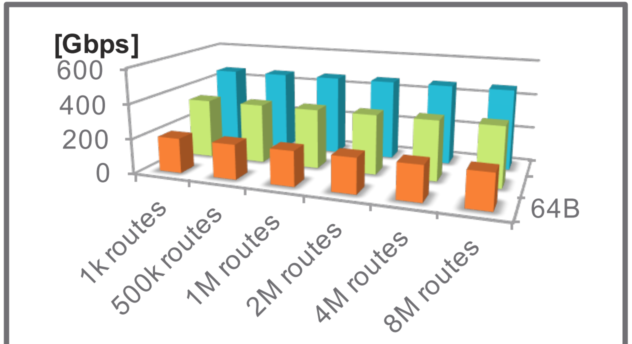


480Gbps zero frame loss

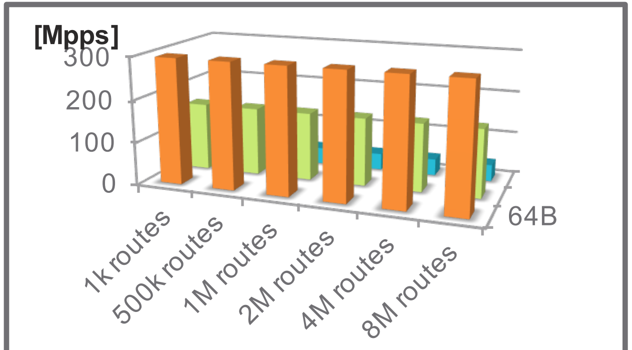


200Mpps zero frame loss

IPv4+ 2k Whitelist, 36 of 72 cores



IMIX => 342 Gbps,1518B => 462 Gbps



64B => 238 Mpps

Phy-VS-Phy

Zero-packet-loss Throughput for 12 port 40GE

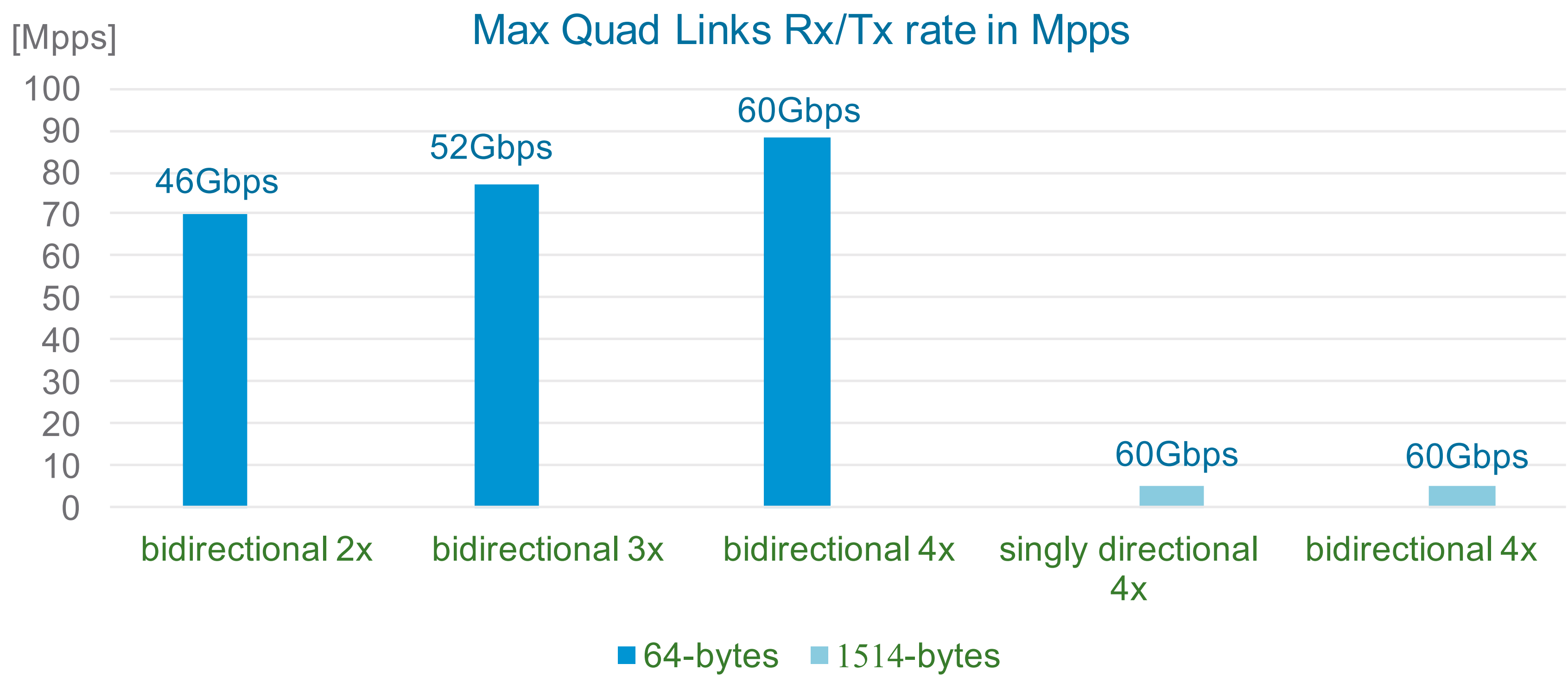
Hardware: Cisco UCS C460 M4
Intel® C610 series chipset
4 x Intel® Xeon® Processor E7-8890 v3 (18 cores, 2.5GHz, 45MB Cache)
2133 MHz, 512 GB Total
9 x 2p 40GE Intel XL710 18 x 40GE = 720GE !!

Latency
18 x 7.7trillion packets soak test
Average latency: <23 usec
Min Latency: 7…10 usec
Max Latency: 3.5 ms

Headroom
Average vector size ~24-27
Max vector size 255
Headroom for much more throughput/features
NIC/PCI bus is the limit not vpp

https://fd.io/resources/

Max Quad Links rx/tx rate



Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz with 20M L3 cache

vSwitching processing complexity

items	spatial complexity	temporal complexity
csp-input	2^12 vlan entries per csp interface	o(1) to find vlan distribution entry
e-line-forward	1 <vlan,interface> entry and 1 <label,nhlfe> entry	all o(1) to find the fwd entries
e-lan-forward	64 <vlan,interface> and 64 <label,nhlfe> and 2^16 fib base entry per e-lan and [n/48,n] fib entry	O(m/48) to find the mac fwd entry, where m is the average hash bucket's list length
cbp-input	2^20 label entries per cbp interface	O(1) to find label distribution entry
pbp-input	2^20 label entries per pbp interface	o(1) to find unicast fwd nhlfe o(n) to enumerate multicast entries where n by default set to 64

Performance expectation

- simpler forwarding logic
- dpdk native context
- burst-oriented and cache optimized and fast index
- expected to scale out with ports across the vSwitch datapath

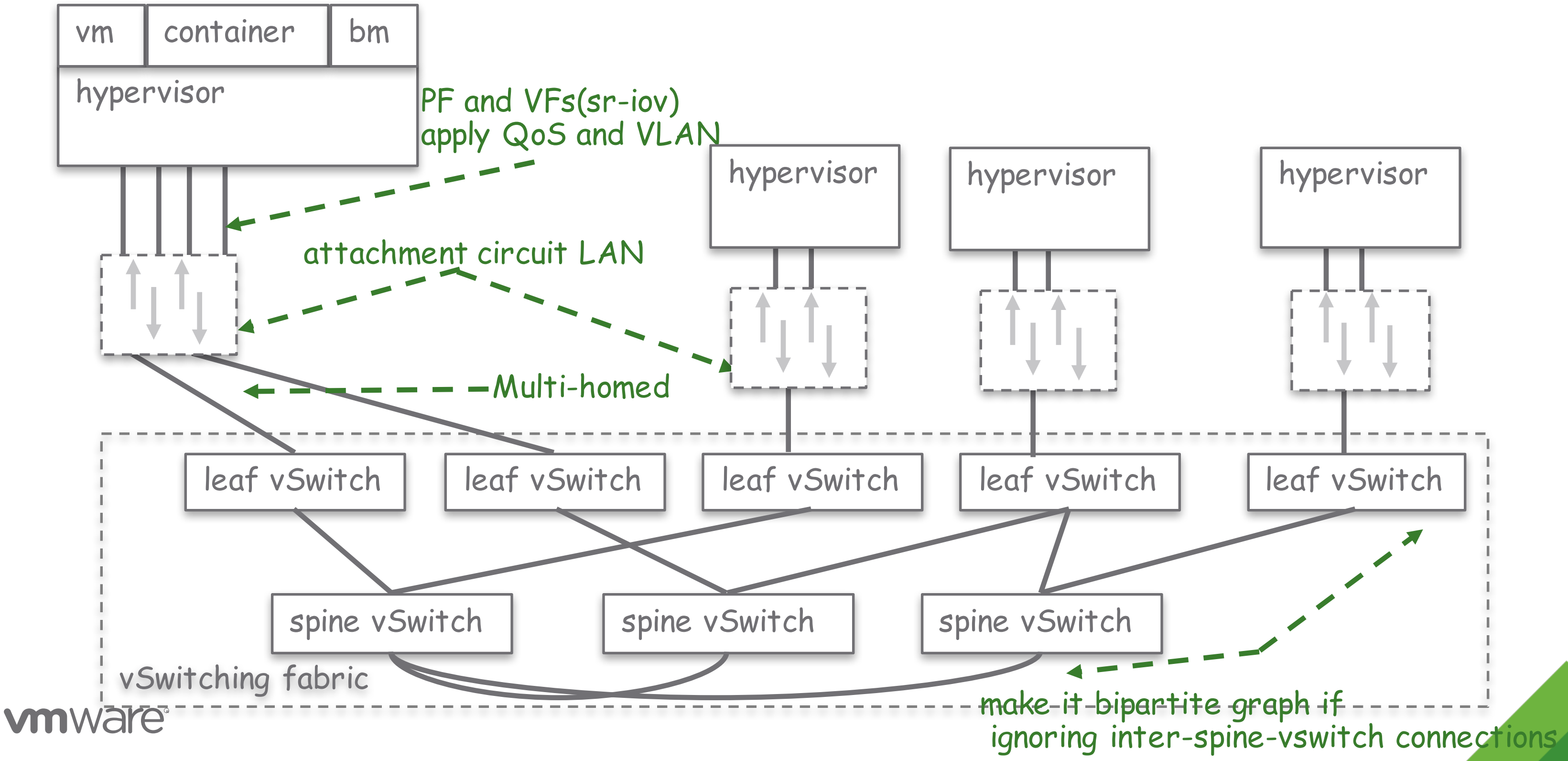
Agenda

- ❑ Overview of Fabric Structure
- ❑ Emulation & Demonstration
- ❑ Performance Evaluation
- ❑ **Use cases Imagination**

Fabric structure review

basic end-system accessing

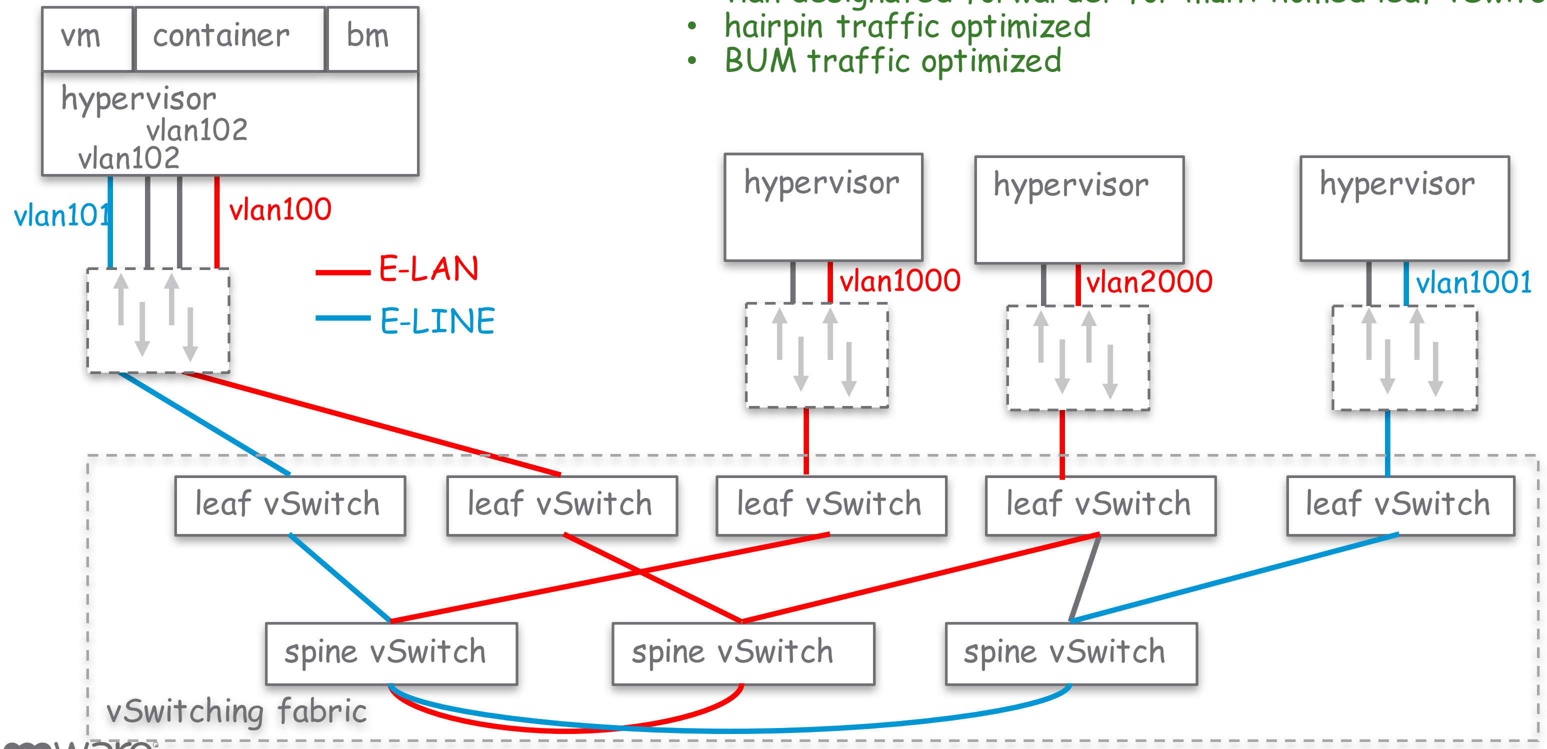
- Try to fully migrate L2 virtual network into infrastructure network domain



Fabric structure review contd

basic end-system accessing

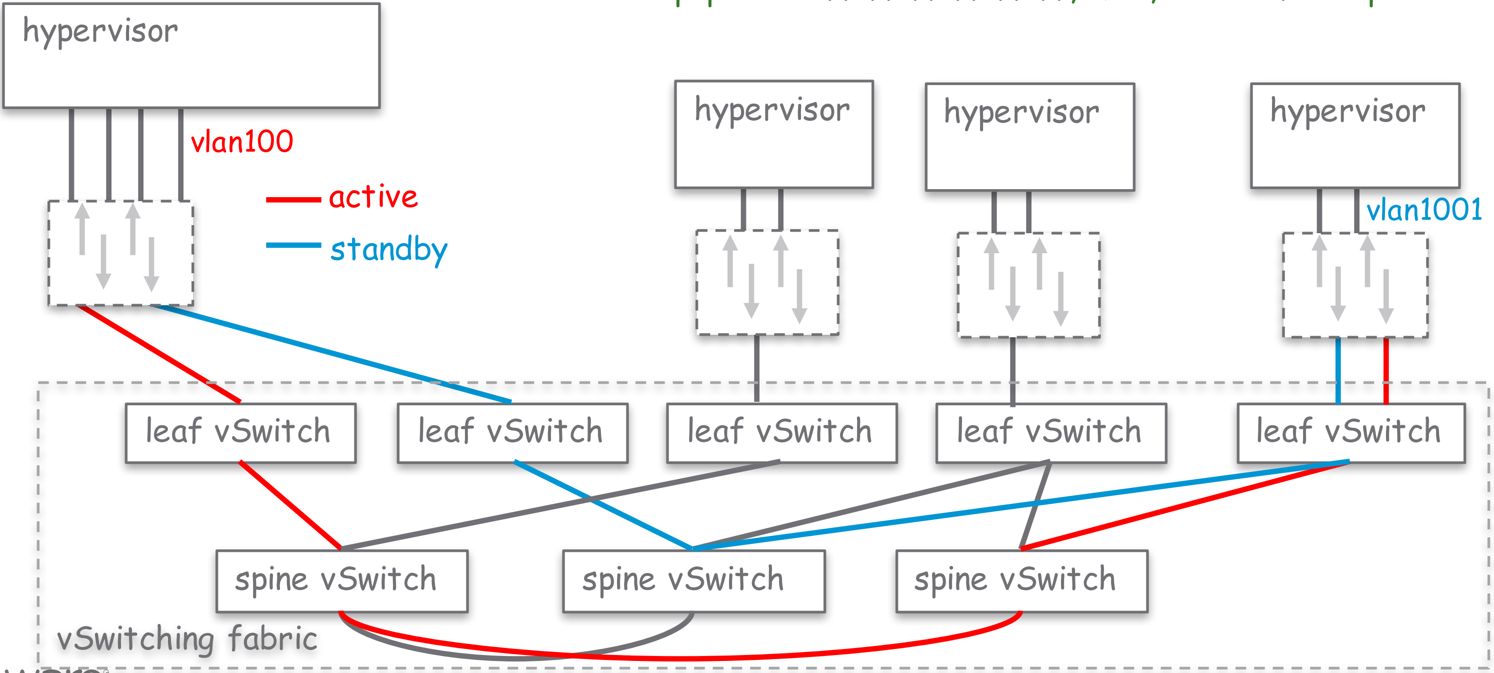
- local vlan scope
- vlan designated forwarder for multi-homed leaf vSwitches
- hairpin traffic optimized
- BUM traffic optimized



Native HA view

Service continuity

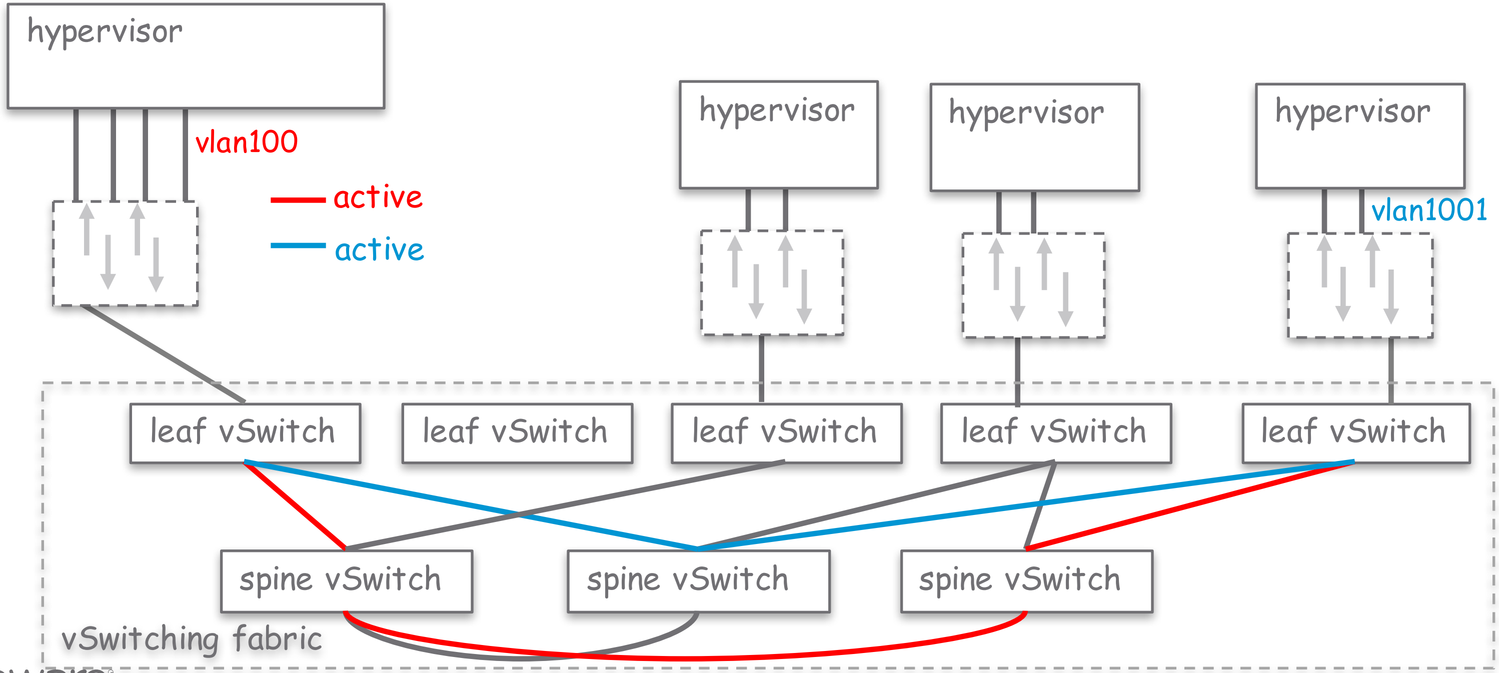
- only active ether-service in forwarding state
- ether-service failure can be detected by in-band OAM or(and) controller
- controller assists ether-service failover
- populate <ff:ff:ff:ff:ff:ff,smac,vlan> to fast update AC lan's mac table



Native ECMP view

service continuity and load balancing

- each ether-service is active
- ether-services are detected by iOAM or(and) controller
- failover can be achieved by disabling inactive ether-service



vmware

More use case features

as a link-level L2 network virtualization solution

- We may still borrow ideas from compute virtualization
  - Snapshot or backup and restore your virtual network
  - Live migration for your virtual network
  - Virtual network High Availability
  - Virtual network fault tolerance (as ECMP?)