Cisco Live!
January 29 - February 2, 2018 • Barcelona
Virtualizing Enterprise Network Functions

Matthias Falkner, Distinguished Engineer, Technical Marketing
Cisco Spark

Questions?
Use Cisco Spark to communicate with the speaker after the session

How
1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install Spark or go directly to the space
4. Enter messages/questions in the space

cs.co/ciscolivebot#BRKCRS-3447
By Way of Introduction …

Matthias is a Distinguished Engineer, Technical Marketing and has been with Cisco for 17 years.

Matthias currently focuses on the evolution of Enterprise and SP network architectures, particularly on end-to-end networking, virtualization and orchestration.

Matthias has held various positions in both Sales and the Business Unit.

I have a strong interest in Cisco DNA and it’s components – a passion I hope to share with you via this presentation!
Network Function Virtualization (NfV) is gaining increasing traction in the industry based on the promise of reducing both CAPEX and OPEX using COTS hardware. This session introduces the use-cases for virtualizing Enterprise network architectures, such as virtualizing branch routers, IWAN deployments, or enabling enterprise hybrid cloud deployments. The sessions also discusses the technology of Virtualization from both a system architecture as well as a network architecture perspective. Particular focus is given on understanding the impact of running routing functions on top of hypervisors, as well as the placement and chaining of network functions. Performance of virtualized functions is also discussed.
Agenda: BRKCRS-3447
Virtualizing Enterprise Network Functions

- Introduction & Motivation
- Virtualization Use-Cases
- Virtualized System Architecture: Building Blocks
- Understanding Performance in a virtualized System
- Secure Agile Exchange: Virtualized connectivity to Multi-cloud Applications
- Summary & Conclusion
Introduction & Motivation
Virtualization of Network Functions (NFV) – Current State

- Idea of de-coupling software from hardware is not new!
- Linked to automation / orchestration
- Increased focus to simplify Enterprise architectures
  - Particularly on L4-7 services
- SPs drive adoption, but Enterprises are following suit
  - Both consumption models (MSP, self-managed) considered
Why Virtualize? Motivations for the Enterprise

CAPEX
• Deploy on standard x86 servers
• Economies of scale
• Service Elasticity – deploy as needed
• Simpler architectural paradigm
  • HA still needed?
• Best-of-breed

OPEX
• Deployment Flexibility
• Reduction of number of network elements
• Reduction of on-site visits
  • Deployment of standard on-premise hardware
  • Simplification of physical network architecture
• Leveraging Virtualization benefits
  • Hardware oversubscription, vMotion, ..
• Increased potential for automated network operations
• Re-alignment of organizational boundaries
Virtualization Use-Cases
The Top 5 use-cases for Enterprise Virtualization
1. Branch Virtualization

- Branch virtualization attractive due to sheer number of branches
- Very different environment to DC!
  - Cost sensitive, often no on-site support, ..
2. Connecting to Apps in the Cloud!

• Enhancing network functionality to reach cloud-based applications

• Support for Amazon AWS, Microsoft Azure, AliCloud etc.
3. Virtualization helps the move to multi-cloud

- Apps increasingly live in multiple clouds
  - Cloud provider Independence
  - SaaS
- CoLo Virtualization facilitates multi-CP connectivity
4. Thin Branches and MSP Virtualization

- Many MSPs are already offering virtualized CPE services
  - AT&T

- Remove complexity (Functions) from the branch
- Leverage virtualization in the SP Data Center / PoP to add services
5. The obvious place to virtualize - Control Plane Functions

- Route Reflectors, PfR MC, LISP MS/MR, WLC…

<table>
<thead>
<tr>
<th></th>
<th>ASR1001 &amp; ASR1002-X (8GB)</th>
<th>ASR1001 &amp; ASR1002-X (16GB)</th>
<th>CSR1000 v (8GB)</th>
<th>CSR1000v (16GB)</th>
<th>RP2 (8GB)</th>
<th>RP2 (16GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipv4 routes</td>
<td>7M</td>
<td>13M</td>
<td>8.5M</td>
<td>24.8M</td>
<td>8M</td>
<td>24M</td>
</tr>
<tr>
<td>vpnv4 routes</td>
<td>6M</td>
<td>12M</td>
<td>8.1M</td>
<td>23.9M</td>
<td>7M</td>
<td>18M</td>
</tr>
<tr>
<td>ipv6 routes</td>
<td>6M</td>
<td>11M</td>
<td>7.4M</td>
<td>21.9M</td>
<td>6M</td>
<td>17M</td>
</tr>
<tr>
<td>vpnv6 routes</td>
<td>6M</td>
<td>11M</td>
<td>7.3M</td>
<td>21.3M</td>
<td>6M</td>
<td>15M</td>
</tr>
<tr>
<td>BGP sessions</td>
<td>4000</td>
<td>4000</td>
<td>4000</td>
<td>4000</td>
<td>8000</td>
<td>8000</td>
</tr>
</tbody>
</table>
Architectural Change with DNA Virtualization

Control & Policy Plane

Service Definition (Design)
- Identity
- Policy Definition
- Catalogue

Orchestration
- Site Ctrl.
- WAN Ctrl.
- DC / Cloud Ctrl.
- Security Ctrl.

Analytics & Assurance
- Ingestion
- Infrastructure Analytics
- Service / Policy Analytics
- Storage

Network Plane

Sites
- Outdoor
- Mobile / 5G
- Branch
- Campus
- Fabric

WAN
- SP
- Internet
- MPLS
- Corporate WAN

Cloud Exchange
- Internet
- VPC
- SaaS

DC Fabric
- Ent Apps

Endpoint Controller
- AD

App Controller

Network Plane

NW Fns (phy&vir)

NW Fns (phy&vir)

NW Fns (phy&vir)
Virtualized System Architecture: Building Blocks
The 4 Layers of a virtualized System Architecture

1. ISR 4000 + UCS E-Series
2. Network Functions Virtualization Infrastructure Software (NFVIS)
3. Virtual Router (ISRv, CSR), Virtual Firewall (ASA, NGFW), Virtual WAN Optimization (vWAAS), Virtual Wireless LAN Controller (vWLC), 3rd Party VNFs
4. Automation / Orchestration (DNA Center, NSO)
A System Architecture View with 3 VNFs

- Example: 3 CSR VMs scheduled on a 2-socket 8-core x86
  - Different CSR footprints shown
- Type 1 Hypervisor
  - No additional Host OS represented
- HV Scheduler algorithm governs how vCPU/IRQ/vNIC/VMKernel processes are allocated to pCPUs
- Note the various schedulers
  - Running ships-in-the-night
A System Architecture View with 3 VNFs

• Example: 3 CSR VMs scheduled on a 2-socket 8-core x86
  – Different CSR footprints shown
• Type 1 Hypervisor
  – No additional Host OS represented
• HV Scheduler algorithm governs how vCPU/IRQ/vNIC/VMKernel processes are allocated to pCPUs
• Note the various schedulers
  – Running ships-in-the-night
A System Architecture View with 3 VNFs

- Example: 3 CSR VMs scheduled on a 2-socket 8-core x86
  - Different CSR footprints shown

- Type 1 Hypervisor
  - No additional Host OS represented

- HV Scheduler algorithm governs how vCPU/IRQ/vNIC/VMKernel processes are allocated to pCPUs

- Note the various schedulers
  - Running ships-in-the-night
A System Architecture View with 3 VNFs

Example: 3 CSR VMs scheduled on a 2-socket 8-core x86
- Different CSR footprints shown

Type 1 Hypervisor
- No additional Host OS represented

HV Scheduler algorithm governs how vCPU/IRQ/vNIC/VMKernel processes are allocated to pCPUs

Note the various schedulers
- Running ships-in-the-night
Freedom of Choice

Traditional Physical Router
Cisco® 4000 Series ISR
- Centralized services
- Fixed integrated services
- Conservative

Physical Router Virtual Services
- 4000 Series ISR + UCS® E-Series
- Upgradable hardware
- Deterministic routing performance

Enterprise NFV
Virtual Router Virtual Services
- Enterprise Network Compute System (ENCS)
- Elastic routing and services
- Router / Server Hybrid

Virtual Router Virtual Services
- UCS C-Series CSP 2100
- Elastic routing and services
- Performance
- Early adopter

Cisco ONE™
- Access to Ongoing Innovation
- License Portability
- Investment Protection
- Access to Ongoing Innovation
- License Portability
- Investment Protection

Conservative
Virtualization Hardware Sweet Spots

- **High Compute Capacity**
  - TDM or Non-Ethernet Interfaces
  - ISR-4K with UCS-F
  - ENCS 5400
  - CSP-2100 UCS C-Series

- **Lower Compute Capacity**
  - Ethernet Only
  - ISR-4K with or without service container

- Options:
  - **ISR-4K with UCS-F**
  - **ENCS 5400**
  - **CSP-2100 UCS C-Series**
Platform Built for Enterprise NFV

Best of Routing & Compute

Complete Virtualized Services

Open for Third Party Services and Apps

Enterprise Network Compute System

ENCS 5100 Series

ENCS 5400 Series
5400 ENCS Internal Networking

- HW offload for VM-VM traffic
- 6 SR-IOV LAN Networks
- High-speed backplane
- VLAN-aware HW switch PoE
- Cellular, T1, DSL, LAN, GE
- Dual-PHY WAN GE or LAN uplink
- Dedicated management ports
- Software switched path
- Lights-out management

**5400 ENCS Platform**

- Data Path
- Control Path
CSP 2100 Appliance for High-end NFV

• RHEL 7.2 based system

• Manage via GUI, CLI (IOX XR synax), REST API, Netconf/Yang

• GUI & REST connections are over HTTPS

• YANG models – used by NSO or other MANO

• Service Image Types: ISO, OVA, QCOW/QCOW2, VMDK, RAW

• Day.0 config file support for services like CSR1000V, ASAv
Virtualization OS: NFVIS is optimized for VNF deployments!

**Network Hypervisor**
- Supports segmentation of virtual networks
- Abstract CPU, memory, and storage resources

**Zero-Touch Deployment**
- Automatic connection to PnP server
- Highly secure connection to the orchestration system
- Easy day-0 provisioning

**Lifecycle Management**
- Provisioning and launch of VNFs
- Failure and recovery monitoring
- Stop and restart services
- Dynamically add and remove services

**Service Chaining**
- Elastic service insertion
- Multiple independent service paths based on applications or user profiles

**Monitoring**
- Netconf Notification
- Host and VM Statistics
- Packet Capture

**Open API**
- Programmable API for service orchestration
- Rest and NETCONF API
Available VNFs from Cisco for Enterprise (Sample)

<table>
<thead>
<tr>
<th>Network Infrastructure</th>
<th>Virtual Router (CSR1Kv)</th>
<th>Virtual Route Reflector (CSR1Kv)</th>
<th>Virtual PE/IP Router (CSR1Kv)</th>
<th>AppNav and AVC (CSR1Kv)</th>
<th>CML/VIRL</th>
<th>DHCP (CSR1Kv)</th>
<th>VXLAN (L2,L3), OTV, VPLS, LISP (CSR1Kv)</th>
<th>Wireless LAN Controller (WLC/MSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Nexus 1000V</td>
<td>Network Analysis Module (NAM)</td>
<td>Wide Area Application Service (WAAS)</td>
<td>IP SLA (CSR1Kv)</td>
<td></td>
<td></td>
<td>VPLS, LISP (CSR1Kv)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Security</th>
<th>Virtual ASA Firewall (ASA)</th>
<th>IPSec and SSL VPN (ASA)</th>
<th>vNGIPS (SourceFire)</th>
<th>NAT (CSR1Kv)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enterprise Network Controller (APIC-EM)</td>
<td>Prime Network Registrar, IP Express</td>
<td>Cisco Prime Infrastructure, Provisioning</td>
<td>Prime Home</td>
</tr>
<tr>
<td></td>
<td>Prime Service Catalog</td>
<td>Prime Performance Manager, Prime Analytics</td>
<td>Intelligent Automation for Cloud (IAC)</td>
<td>Prime Fulfillment, Order Fulfillment</td>
</tr>
<tr>
<td></td>
<td>Cisco VDS-IS</td>
<td>Unified Contact Center, CC Express</td>
<td>Unified Contact Center, CC Express</td>
<td>Prime Collaboration</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Management &amp; Orchestration</th>
<th>Prime Collaboration</th>
<th>UCS Director</th>
<th>Prime Network Service Controller</th>
<th>Video Conferencing (MSEBK)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prime Access Registrar</td>
<td>Prime Collaboration</td>
<td>Prime Fulfillment, Order Fulfillment</td>
<td>Unified Contact Center, CC Express</td>
</tr>
<tr>
<td></td>
<td>TR payroll (CSR1Kv)</td>
<td>Prime Collaboration</td>
<td>Prime Fulfillment, Order Fulfillment</td>
<td>Unified Contact Center, CC Express</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Voice &amp; Video</th>
<th>CUBE (CSR1Kv) Roadmap</th>
<th>Video Conferencing (MSEBK)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prime Collaboration</td>
<td>Prime Collaboration</td>
</tr>
<tr>
<td></td>
<td>Prime Collaboration</td>
<td>Prime Collaboration</td>
</tr>
<tr>
<td></td>
<td>Prime Collaboration</td>
<td>Prime Collaboration</td>
</tr>
</tbody>
</table>

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public
NFV Automation and Management Options

Network Functions Virtualization Infrastructure Software (NFVIS)

X86 Host
Demo: DNA Center ENFV Design
NFVIS Local GUI
NFVIS Local GUI
### API-Driven NFV Automation: REST Example

**VM deployment**

<table>
<thead>
<tr>
<th>Description</th>
<th>Method</th>
<th>Payload</th>
<th>API</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deploy the VM</td>
<td>POST</td>
<td>yes</td>
<td>/api/config/vm_lifecycle/tenants/tenant/admin/deployments</td>
</tr>
</tbody>
</table>

```shell
curl -k -v -u admin:admin -H Accept:application/vnd.yang.data+xml -H Content-Type:application/vnd.yang.data+xml -X POST https://172.18.73.192/api/config/vm_lifecycle/tenants/tenant/admin/deployments --data <deployment><name>ISRv_small</name><vm_group><name>ISRv</name><image>ISRv.tar.gz</image><flavor><profile-1><flavor><bootup_time>600</bootup_time><recovery_wait_time>0</recovery_wait_time><recovery_policy><action_on_recovery>REBOOT_ONLY</action_on_recovery><recovery_policy><action_on_recovery>REBOOT_ONLY</action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_recovery><action_on_re
Understanding Performance in a virtualized System
Main Trade-off and Research Areas

1. Cost of Virtualization solution as a function of performance

2. Trading-off performance for virtualization flexibility
   • Tuning performance may impact virtualization elasticity

3. Architectural Considerations
   • Capacity planning Service Function Chains?
   • Orchestration solution?
   • High-Availability requirements?
Performance Aspects for VNF Deployments

- Intra-VM Bottleneck
- Hypervisor Bottleneck
- vSwitch Bottleneck
- pNIC Bottleneck
What is the Benchmark to measure against?

- VM throughput Benchmark: ideal throughput that can be achieved with an unconstrained I/O path
  - Derived from
    - Measuring the throughput of a single CSR 1000v with SR-IOV
      - SR-IOV used as best ‘ideal’ I/O path
      - Hypervisor with a single VNF is not stressed – only has to deal with one VNF
    - Assuming perfect additivity for each VNF added
      - Under ideal conditions, each VNF adds the same throughput as the first VNF

Slope depends of feature set & packet Size

Measured Throughput for VM₁

Throughput from VM₁ Multiplied by number of VMs

Benchmark

<table>
<thead>
<tr>
<th>Axis Title</th>
<th>Benchmark</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.819</td>
</tr>
<tr>
<td>2</td>
<td>11.638</td>
</tr>
<tr>
<td>3</td>
<td>17.457</td>
</tr>
<tr>
<td>4</td>
<td>23.276</td>
</tr>
<tr>
<td>5</td>
<td>29.095</td>
</tr>
<tr>
<td>6</td>
<td>34.914</td>
</tr>
<tr>
<td>7</td>
<td>40.733</td>
</tr>
<tr>
<td>8</td>
<td>46.552</td>
</tr>
</tbody>
</table>
Physical System Bottlenecks: Interface and Cores

- Throughput can never exceed the physical interface capacity
- Number of VNFs is constrained by number of cores
  - Need to account for CPU cores required for system Linux
  - Need to account for CPU cores for switching (I/O tax)

<table>
<thead>
<tr>
<th>System Throughput (Gbps)</th>
<th>Number of VNFs</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.819</td>
<td>1</td>
</tr>
<tr>
<td>11.638</td>
<td>2</td>
</tr>
<tr>
<td>17.457</td>
<td>3</td>
</tr>
<tr>
<td>23.276</td>
<td>4</td>
</tr>
<tr>
<td>29.095</td>
<td>5</td>
</tr>
<tr>
<td>34.914</td>
<td>6</td>
</tr>
<tr>
<td>40.733</td>
<td>7</td>
</tr>
<tr>
<td>46.552</td>
<td>8</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Physical Interface Limit: 4x10GE

Max number of VNFs per system:
Number of cores = 2 sockets * 12 cores per socket = 24
Minus: Number of cores for System = 24-2 = 22
Divided by: Number of vCPUs / VNF = 22 / 2 vCPU = 11
The I/O Path is often the key bottleneck in NFV

OVS-DPDK / VPP
- Moderate Performance (4-5 Gbps per system)
- Limited counters

VEB
- High Performance
- Requires NIC support
- BCAST???

VEPA
- High Performance
- Full switching counters
- Requires support in TOR

* Workaround: assign different VLANs to VNFs, and route in TOR
Hypervisor Traversal Tax: Example KVM with OVS

- KVM with OVS consumes a vHost thread per configured VM interface
- The vHost thread is very CPU intensive, requires dedicated physical core
- On 16-core server, can only get 3 CSR1000v (2vCPU, 2 i/f each)
  - Cores for CSR: 6
  - Cores for vSwitch: 2
  - Cores for vHost: 6
  - Free: 2
- Should be considered when service chaining

Hypervisor traversal tax = 8/16 = 50%
May not be fully utilized!
Virtualizing I/O – Sample System Architectures
Virtualizing I/O – Sample System Architectures
Packet Path from Physical Interface into VNF

Why does this matter?
- Illustrates contention of shared resources
- Each packet move consumes resources
- Packet pointer buffers have limited depth
  - can cause drops

Packet feature processing
Packet moved to VNF
VNF interrupted, packet pointer passed to buffer
vHost-user notified
VPP kicked, switching packet
Packet copied into Memory
Packet Arrival
Packet feature processing
Packet moved to VNF
VNF interrupted, packet pointer passed to buffer
vHost-user notified
VPP kicked, switching packet
Packet copied into Memory
Packet Arrival
The Hypervisor can impact Performance!
SR-IOV, FD.io and OVS-DPDK with CEF

Multi-VM Throughput (Gbps) with various I/O architectures
IPv4 Forwarding, IMIX Packet Size, XE 16.3.1

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best case Additive (CEF)</td>
<td>7.4</td>
<td>14.8</td>
<td>22.2</td>
<td>29.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OVS-DPDK (CEF)</td>
<td>5.6</td>
<td>8.6</td>
<td>12.2</td>
<td>11.1</td>
<td>10.3</td>
<td>8.6</td>
<td>8.6</td>
<td>7.7</td>
<td>7.5</td>
<td>4.6</td>
</tr>
<tr>
<td>SR-IOV (CEF)</td>
<td>7.4</td>
<td>14.3</td>
<td>19.017</td>
<td>19.0</td>
<td>19.0</td>
<td>19.0</td>
<td>19.0</td>
<td>19.0</td>
<td>19.0</td>
<td>19.0</td>
</tr>
<tr>
<td>FD.io VPP (CEF)</td>
<td>5.9</td>
<td>11.5</td>
<td>10.6</td>
<td>10.2</td>
<td>10.1</td>
<td>9.7</td>
<td>8.9</td>
<td>8.5</td>
<td>8.9</td>
<td>8.6</td>
</tr>
</tbody>
</table>
Feature Configurations require CPU Cycles!
FD.io performs well with large packets
Secure Agile Exchange: Virtualized connectivity to Multi-cloud Applications
Secure Agile Exchange – Virtualized secure Cloud on-ramp
Secure Agile Exchange – Virtualized secure Cloud on-ramp
Secure Agile Exchange – Virtualized secure Cloud on-ramp
Secure Agile Exchange – Virtualized secure Cloud on-ramp
Secure Agile Exchange – Virtualized secure Cloud on-ramp

Customers

Employees

Partners

Enterprise Onramp

Private Data Center

Cloud
Secure Agile Exchange – Virtualized secure Cloud on-ramp
Secure Agile Exchange – Virtualized secure Cloud on-ramp
Secure Agile Exchange – Virtualized secure Cloud on-ramp
Secure Agile Exchange – Virtualized secure Cloud on-ramp
Secure Agile Exchange – Virtualized secure Cloud on-ramp

- Customers
- Employees
- Partners

Colocation Centers

Private Data Center

Cloud
Secure Agile Exchange – Virtualized secure Cloud on-ramp
Secure Agile Exchange – Virtualized secure Cloud on-ramp
Secure Agile Exchange – Virtualized secure Cloud on-ramp
SAE - Let's start virtualizing the DMZ

DMZ (physical)

Enterprises Network

pFW  SLB  pFW  Web  pFW  ALG  pFW

APP  DB

DMZ (virtualized)

Enterprises Network

pFW  SLB  vFW  vFW  vFW  vFW  vFW  vFW  vFW  vFW  vFW  vFW  vFW  pFW

SLB  vFW  vFW  vFW  vFW  vFW  vFW  vFW  vFW  vFW  vFW  vFW  vFW  pFW

APP  DB
Top 3 Reasons to Virtualize the DMZ

1. Per-App network functions and operation

2. Dynamic & automated service insertion with focus on Security

3. Usage-based consumption model
But wait – SAE is MORE than a virtualized DMZ

• Applications are moving into the cloud!
But wait – SAE is MORE than a virtualized DMZ

• Applications are moving into the cloud!
But wait – SAE is MORE than a virtualized DMZ

• Applications are moving into the cloud!
But wait – SAE is MORE than a virtualized DMZ

• Applications are moving into the cloud!
But wait – SAE is MORE than a virtualized DMZ

- Applications are moving into the cloud!
Top 4 Reasons for CNF / CoLos

- **Simplicity**
  - Automation
  - Orchestration
  - Hardware reduction

- **Fidelity**
  - Carrier / CP Independence
  - Tight SLA

- **Economics**
  - Regional CoLo Breakouts
  - Reduction of Circuit Costs

- **Security**
  - Consistent Security
  - Central interconnection point
Addition Policy to the Mix

- Expressing the relationships between Employees, customers, partners and applications by policy
Addition Policy to the Mix

• Expressing the relationships between Employees, customers, partners and applications by policy
Powerful Components

- **CSP 2100**
  - VNF Hypervisor
  - Performance Focused Hardware
  - CLI, GUI, and API Driven

- **Cisco & 3rd Party VNFs**
  - Virtual First Focus
  - Consistent Software between Virtual and Hardware

- **Flexible Switching Fabric**
  - VNF Ready Fabric
  - Scales from Small to Large Deployments

- **Open Orchestration**
  - Automate and Orchestrate Cisco and 3rd Party VNFs
  - Create repeatable service chain models
SAE Building Blocks

- Border Elements: Routers, Firewall
- L2/L3 fabric / Switching
- X86 hosts for service chaining  
  - Hosting VNFs
- Orchestration
- Analytics
- Service Catalog (Optional)  
  - Templated service chains
SAE Service Chains / Virtualization

- X86 hosts to provide virtualized services
  - Virtual routing, firewall, loadbalancing, web-filter, remote-access, …
  - Support for Cisco and 3rd party
  - Supported VNFs are in a VNF catalogue

- Service chaining support
  - Initially configured via TOR based on SR-IOV
  - Other options under investigation
Typical SAE Consumer / Provider VNF Service Chains

Employee

- WAN Branch
- vEdgeCloud
- Service insertion on router WAN interface
- WAAS Node
- CSR with AppNav-XE

Public Cloud

- Internet
- Cisco Secure Agile Exchange
- HTTP, HTTPS, FTP, FTPS
- Redirect from ASA via WCCP
- Redirect from WSA via Internet
- Content Adaption Protocol (ICAP)
- Direct Connect
- Cloud

Partner

- Partner Router
- VPN Tunnel
- ASA
- Firepower (FTDv)

Data Center

- ASA
- Firepower (FTDv)
- CSR
- WSA
- Collocation
- Digital Guardian
- Cloud

Cisco Secure Agile Exchange
Summary & Conclusion
Agenda: BRKCRS-3447
Virtualizing Enterprise Network Functions

- Introduction & Motivation
- Virtualization Use-Cases
- Virtualized System Architecture: Building Blocks
- Understanding Performance in a virtualized System
- Secure Agile Exchange: Virtualized connectivity to Multi-cloud Applications
- Summary & Conclusion
Key Takeaways

• Network Function Virtualization is a standard tool in the tool-box of network engineers

• Key benefits:
  • Operational simplicity - deploy functionality within minutes
  • Leverage power of programmability
  • Potential to achieve architectural simplification

• Cisco Virtualized Network Functions offer full feature richness and consistency with their hardware variants

• Important to understand the system architecture, in particular with a view to performance

• Cisco is offering full-stack solutions to simplify NFV Deployments (e.g. ENFV, SAE)
Related Cisco Live Barcelona 2018 Sessions

- LTRVIR-2100 CSR1k public cloud lab
- BRKARC-2014 Branch Virtualization – The Evolving NFV Landscape
- BRKARC-2112
- Deploy Network Services in Minutes on any Platform with Cisco ENFV
- BRKRST-1888
- Routing Underlay and NFV Automation with DNA Center
- BRKRST-2557
- Leveraging NSO for SD-WAN, SDA, and ENFV
- TECSPG-2300
- Network Function Virtualization Seminar
ENFV Resources


Selected References on VNF Performance


- CSR 1000v in Microsoft Azure Deployment Guide

- CSR 1000v Unified Performance Testing Results

- EANTC OVS and VPP Performance report
  - With significant input from the CTAO team under D. Ward!
  - Youtube Video: [https://www.youtube.com/watch?v=Z5M0Zl0uvj0](https://www.youtube.com/watch?v=Z5M0Zl0uvj0)

- Vnet SLA Performance Testing (Cisco Intern only)
  - [http://wikicentral.cisco.com/display/PACKETCOM/Virtual+Network+SLA](http://wikicentral.cisco.com/display/PACKETCOM/Virtual+Network+SLA)

- OVS Performance
  - [OVS Performance Characterization (Madhu Challa)](http://wikicentral.cisco.com/display/PACKETCOM/Virtual+Network+SLA)
  - [OVS Performance Blog](http://wikicentral.cisco.com/display/PACKETCOM/Virtual+Network+SLA)
  - [OVS Performance Characterization Paper, TU Munich, IEEE Conference on Cloud](http://wikicentral.cisco.com/display/PACKETCOM/Virtual+Network+SLA)
Performance Analysis of Virtualized Network Functions on Virtualized Systems Architectures

Matthias Falkner¹, Aria Leivadeas¹, Iasonas Lambadaris¹ and George Kesidis²
¹Technical Marketing, Enterprise Infrastructure and Industry Group, Cisco Systems, Email: mfalkner@cisco.com
²Department of Systems and Computer Engineering, Carleton University, Email: georgek@carleton.ca

Abstract—Network Function Virtualization (NFV) is an emerging area of research that aims at transforming the concept of virtualization and allows the consolidation of many network services on top of an industry standard off-the-shelf computing platform. NFV aims at reducing expenses compared to providing services from dedicated and expensive hardware appliances and has led to Enterprises and Service Providers to increasingly make use of Virtualized Network Functions (VNFs) to reap the benefits of reduced capital and operational expenses. Total cost of ownership calculations however are typically a function of the available network throughput and performance which is a virtualized system highly dependent on the overall system architecture. The authors of [1] analyze the characteristics of the underlying hypervisor solution, the path from the hypervisor to the VNFs are examples of how the system architecture can influence overall performance and throughput. This article provides the challenges of deploying VNFs on a virtualized system architecture and analyses the impact of the architecture on the overall VNF performance under both single-VNF and multi-VNF configurations.

I. INTRODUCTION

During the last decade, an explosive increase of new services and services has been noticed. However, the need to launch new network services and applications is often associated with the requirement to develop, deploy and install new hardware appliances in the existing infrastructure. This entails an inflow of the cost for acquiring and using dedicated hardware appliances by the service providers, while users are experiencing a remote increase in the service price.

Network Function Virtualization (NFV) is an emerging network paradigm that proposes to turn these dedi- cated middleboxes into software-based virtualized net- work functions called Virtualized Network Functions (VNFs) [1]. In particular, NFV leverage virtualization technology to consolidate many network equipment types onto commercial off-the-shelf (COTS) hardware. Hence, the high specialized equipment can be replaced by industry standard servers that can respond better in a dynamic changing environment and deploy new services as needed. This service providers are increasingly making use of VNFs to reduce the overall capital expenses (CAPEX). Furthermore, operators are also reporting a significant reduction in the operating expenses (OPEX) by increasing the level of automation enabled.

Specially, NFV is commonly to be seen in configuration with other emerging network technologies such as Software Defined Networking (SDN) and Cloud Computing. SDN along with NFV can provide more efficient traffic steering (alleviating the impact of flow between the VNFs). Similarly, NFV can be complemented with technologies developed for cloud computing. The opera- tional overlap between NFV and Cloud Computing is high as the core of both technologies in virtualization. Cloud enables the execution of software-based work- loads, such as VNFs, as a shared pool of resources, while it allows the elastic scaling of the allocated resources according to the demands of the VNFs.

Total cost of ownership calculations however are typi- cally a function of the available network throughput and performance on a virtualized system. The computational resources, the number of VNFs running on a VM, their resource demands, and the capabilities of the system are factors that can influence the overall performance and throughput of the system. This paper provides insights into these factors that affect the performance of VNFs that are hosted in virtualized system architectures.

Specifically, this paper provides test results from a Kentik Virtual Machine (KVM) [2] hypervisor environmen- t, using a CSR 1000v virtual Cisco router.

The remainder of the paper is structured as follows: Section II provides a overview of allocating and man- aging VNFs on top of COTS server. Section III describes the overall system architecture of running VNFs in a generic hypervisor environment on top of COTS server. Following, Section IV assesses the performance of running VNFs on top of standard industry server. Finally, Section V concludes the paper.
Cisco Spark

Questions?
Use Cisco Spark to communicate with the speaker after the session

How
1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install Spark or go directly to the space
4. Enter messages/questions in the space

cs.co/ciscolivebot#BRKCRS-3447
• Please complete your Online Session Evaluations after each session

• Complete 4 Session Evaluations & the Overall Conference Evaluation (available from Thursday) to receive your Cisco Live T-shirt

• All surveys can be completed via the Cisco Live Mobile App or the Communication Stations

Don’t forget: Cisco Live sessions will be available for viewing on-demand after the event at www.ciscolive.com/global/on-demand-library/.
Continue Your Education

• Demos in the Cisco campus
• Walk-in Self-Paced Labs
• Tech Circle
• Meet the Engineer 1:1 meetings
• Related sessions
Thank you