Network Function Virtualization (NFV) using IOS-XR

Syed Hassan
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#BRKSPG-2724
Agenda

- Role of NFV in Network
- IOS XRv 9000
- Benefits & Use Cases
- Deployment & Troubleshooting
- Service Orchestration for NFV
- Summary
Network Functions Virtualization (NFV) - Defined

.... NFV decouples the network functions such as NAT, IPS, DNS, RR etc. from proprietary hardware appliances, so they can run in software.

Network Functions Virtualization (NFV) - Defined

It utilizes standard IT virtualization technologies that run on high-volume service, switch and storage hardware to virtualize network functions.

Network Functions Virtualization (NFV) – Market Drivers

Traditional Network
Separate appliance for each function

- **Proprietary Software**
  - Designed to run on custom hardware

- **Proprietary Hardware**
  - Custom FPGA/ASIC/Optics/CPU …

- **Fixed Network Function**

- **Limited Scalability:**
  - Physical Space and Power limitations

- FW
- PE
- CE
- NA
- T/G
- BNG
- EPC

NFV
Virtualized Function on High Capacity Device

- **Proprietary Software with open APIs**
  - Designed to run on **generic** hardware

- **Generic Hardware:**
  - **Standard** FPGA/ASIC/Optics/CPU …

- **Flexible Network Function**

- **Cloud Scale**

- vFW
- vNA
- vBNG
- vPE
- vCE
- vEPC

**Network infrastructure/Service Functions** run on **Virtualized** compute platforms.
Network Functions Virtualization (NFV) – Market Ask

- **Deliver Agility & Flexibility**: 86%
  - Scale up or down services to address changing demands

- **Opex Optimization**: 59%
  - Reduce space, power and cooling requirements

- **Faster Time to Market**: 69%
  - Easy trial & deployment of new services

- **Reduced Capex**: 62%
  - Reduce purpose-built HW; Rely on Industry standard NFVI

Based on Survey Results by SDX Central
Enablers of SP Network Transformation

**NFV**
- Agility & Flexibility
- Operational Efficiency
- Hardware Agnostic

**SDN**
- Automated Provisioning
- Traffic Optimization
- Service Agility
- Reduced Cost

Network Functions Virtualization (NFV)

Application & Service Orchestration

Software Defined Networking (SDN)
Network Functions Virtualization & SDN

- Orthogonal Concepts; Similar Goals
- SDN complementary, but not mandatory
Network Functions Virtualization & SDN

Comparison

<table>
<thead>
<tr>
<th>SDN</th>
<th>NFV</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDN Controller: Open Daylight (ODL), XR Transport Controller, Wan Automation Engine (WAE) etc.</td>
<td>Virtual network functions: vFW, vRR, vCPE, vPE etc.</td>
</tr>
<tr>
<td>OpenFlow, NETCONF/Yang, Path computation element protocol (PCEP)</td>
<td>VM to Host (socket, Taps etc.)</td>
</tr>
<tr>
<td>Involves end to end networking</td>
<td>Involves single network entity</td>
</tr>
<tr>
<td>New network architecture</td>
<td>Virtualization of existing architecture</td>
</tr>
</tbody>
</table>
Achieving Network Function Virtualization

- Applications and running using virtualized Hardware end CPUs
- Guest O/S running independently in each VM
- HyperVisor - isolated application providing VMs on the Host
- Basic host operating system
- Virtualization capable CPUs
Network Virtualization
ETSI Architecture Framework for NFV
Virtualizing Network Functions

X86 versus Custom Network Processing Unit (NPU)

**Network Forwarding (L0-3)**
- Mostly predictable traffic
- Stateless functions
- High throughput / BW
- IPv6/v4, MPLS, VPNs, Optical

**Network Services (L4+)**
- Unpredictable traffic
- Stateful functions
- Low to Med Throughput
- BGP Route reflector, Firewall, DPI

Better Fit for Custom Hardware

Better Fit for NFV

- Bandwidth
- Compute
NFV across Cisco portfolio
Virtualized Network Operating Systems

**IOS-XR**
Virtualized as IOS XRv 9000

**NX-OS**
Virtualized as Nexus 1000v

**IOS-XE**
Virtualized as CSR1000v

**ASA**
Virtualized as ASAv
Cisco’s VNF Portfolio ....

IOS XRv 9000
CSR1000v
DDoS Scrubber (w/Arbor)
Nexus 1000v
ASA v
QvPC
vNAM
vWSA
vWAAS
vWLC
vWSA
vESA
Network Transformation with NFV
Network Transformation with NFV

Diagram showing network components labeled as CE, Access, PE/Agg, Core, RR, Contr, and VNF.
Network Transformation with NFV
IOS XRv 9000
IOS XR

- Time *hardened* for years
  - CRS-1, CRS-3, CRS-X, ASR 9000, NCS 6000
- High-scale control plane
- MicroKernel-based
- Modular Software
- Process Restartability & Redundancy
- Remediation through add-on patches
**IOS XRv**

- IOS XR on x86 **Virtualized** environment
- Full *Platform Independent* IOS XR
  - Same IOS XR software feature set
  - Manageability
  - Control Plane
  - Routing
EOS XRv - One Physical hardware -- Multiple Instances
IOS XRv 9000

- Virtualized IOS XR with Control and Data plane Separation
  - Linux Containers for Admin, Control and Data Planes
  - 64 Bit Kernel

- Scalability through Flexible resource Allocation
  - Data plane scalability.
  - Control Plane scalability
IOS XRv 9000

Routing & Management Plane
XR Route Processor Functionality
XR Line Card Functionality
Support for Physical & Virtual Data-Plane

Forwarding Plane
Virtual Forwarder
Software Based H/W assist
Common code base as physical ASICs

Admin Plane
Infra management
SMU management
VM/LXC Lifecycle Management
Upgrade/Downgrade
Light Weight

IOS XRv 9000
HyperVisor
Host

L3FIB  QoS  L2FIB  ACL

MTRIE  Policer  Intf
IOS XRv 9000

Right sizing Scale and Throughput through Control and Data Planes

N x LCs (NPU) : 1 x RP (CPU)

Routers + LCs

IOS XR

RP (Control Plane)

LC (Data Plane)

N x NPU: M x CPU

Virtual Routers

Routers/Compute

LC (Data Plane)

LC (Data Plane)

LC (Data Plane)

Compute

Compute Server (Control Plane)

Compute Server (Control Plane)

Compute Server (Control Plane)
IOS XRv 9000

Architecture

- XRV9K – Control Plane
- XRV9K Libraries & Binaries
- VPP / FD.IO Libraries and Binaries
- DPDK
- Physical/Logical Interface; SR-IOV etc.

VPP: Vector Packet Processing
DPDK: Data Plane Development Kit
IOS XRv 9000
VPP & DPDK Interaction

Graph & Batching of Packets
Send/receive batches of packets to the dpdk rx burst and tx burst functions & Statically links to DPDK

VPP-Vnet (TX/RX)  VPP-Vlib (CLI/Graph)  VPP-Executable (Core VPP)  VPP-Infra (Low level)

VPP / FD.IO Libraries & Executable

Heap memory management, timers, an vector operation abstraction

Physical/Logical Interface ; SR-IOV etc.

XRv9K – Data Forwarding & Features (L3/L2 Forwarding, Feature - qos/acl/pbr/1I etc)
XRv9K – (DPA/HAL)

XRv9K – Control Plane

XRv9K Libraries & Functions
Cisco IOS XRv 9000
NFV Design Trade Offs
IOS XRv 9000 Positioning
Completing the XR Edge Portfolio
Benefits & Use Cases
IOS XRv 9000

Benefits

Lower Opex
Easy provisioning, configuration & deployment for VMs

Green
low power consumption → Lower carbon footprint

Flexible Design
CP & DP Separation and independent resource allocation

Elastic
Dynamic resource allocation & de-allocation

Lower Capex
IOS XRv on standard compute
Consumption Based Growth

SDN Ready
Independent control and forwarding
IOS XRv 9000
Use Cases

- Education and Training
- Network Simulation & Planning
- Network Functions Virtualization

Universities – Enterprise – Public Sector – Service Providers
IOS XRv 9000 VNF Use Cases
IOS XRv 9000 as vRR

- Traditional Role of RR
  - BGP peering
  - Solve N*N full-mesh BGP interconnect
  - Distribute BGP routes to PEs

- Not in packet path
- Focal point for iBGP sessions
IOS XRv 9000 as vRR

RR role evolution - centralized provision, services, and applications

Per Service
Per Address Family
Redundant
Optimized Placement
Scalable
Easy Provisioning
IOS XRv 9000 as vRR

Performance (Multi-Core)
Independent Operation
High Availability
Same BGP Implementation (XR)

Without Compromising

Primary Backup
IPv4 RR
Vpnv4 RR
IPv6 RR
L2vpn vRR

Primary Backup
IPv4 RR
Vpnv4 RR
IPv6 RR
L2vpn RR

8 Physical Devices

2 Physical Devices
Virtualized RRs per AFI
IOS XRv 9000 as vPE

Forwarding Performance (Multi-Core)
Consumption Based Growth
Control Plane & High-Performance Data Plane
High Availability

L3VPN
Customer A
L3VPN
Customer B
L3VPN
Customer C
IOS XRv 9000 as SDN Controller

- IOS XRv 9000
  as: XR Traffic Controller (XTC)

- WAE Application
- NSO

- Core ISIS L2
- DC ISIS L1
- Access ISIS L1
IOS XRv 9000 as a vBNG

Data Center

vBNG vBNG vBNG vBNG

IP/MPLS Network

DCI DCI

INET GW WEB

IOS XRv 9000
Virtualizing Network Functions using IOS XRv 9000

- Performance (Multi-Core)
- Independent Operation
- High Availability
- IOS XR Based Implementation
- Elasticity & Flexibility
- Portability & Agility
- Route Scalability (64b OS)
- Management & Orchestration
- Lower Opex/Capex
## Virtualizing Network Functions using IOS XRv 9000

<table>
<thead>
<tr>
<th>Physical Router</th>
<th>VRR on UCS Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Power consumption ~425W</td>
<td>Max Power consumption ~410W</td>
</tr>
<tr>
<td>Total power for 8 instance ~3.4kW</td>
<td>Total power for 8 instances ~820W</td>
</tr>
<tr>
<td>Power/Year = 29,785 KWh</td>
<td>Power/Year = 7,182 KWh</td>
</tr>
<tr>
<td>Power Cost/Year = $5,659 (19c/kWh)</td>
<td>Power Cost/Year = $1,364 (19c/KWh)</td>
</tr>
</tbody>
</table>

Power Calculations based on ASR9001 (Max Power)
UCS C240 M3 SFF with Intel E5-2643 v2 3.30 GHz/130W 6C/25MB Cache/DDR3 1866MHz with 96 GB Mem, 4 HDD with RAID, and 1 Adapters.
Physical Network Device vs NFV
Consumption Based Deployment

Physical Network Device
- Under-Provisioned
- Over-Provisioned

Network Function Virtualization
- Consumption based capacity growth

CPU, Memory, Gbps
Time
Capacity Demand
Capacity Deployed
IOS XRv 9000 Deployment
IOS XRv 9000 Hardware/Software Requirements

• Hardware
  • Any x86-based server capable of virtualization
    • e.g. Intel® CPUs with VT-x support

• Hypervisor
  • hypervisor agnostic
  • VMWare ESXi 5.5/6.0, QEMU/KVM (RHEL & CentOS 7,7.1/7.2/7.3, Ubuntu 14.04.03/16.04 LTS)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Minimum</th>
<th>Recommended/Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU (Cores)</td>
<td>2 (1 Control Plane, 1 Data Plane)</td>
<td>4 Sockets</td>
</tr>
<tr>
<td>Memory (RAM)</td>
<td>12 GB</td>
<td>19GB recommended</td>
</tr>
<tr>
<td>Hard Disk</td>
<td>45GB</td>
<td></td>
</tr>
<tr>
<td>NIC Port</td>
<td>4</td>
<td>Production 11 (2 reserved, 8 traffic)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Simulation : 128</td>
</tr>
<tr>
<td></td>
<td>(E1000, VirtIO, Intel1/10/40G, VMXNet3, SR-IOV)</td>
<td></td>
</tr>
</tbody>
</table>
IOS XRv 9000 Features

- MP-BGP/eBGP, BGP 3107, FlowSpec
- OSPF/ISIS etc.
- **BFD** (Bidirectional Forwarding Detection)
- LDP/MPLS, 6PE, 6vPE, RFC 3107 (3 labels), L3VPN
- IPv4 ACL (chained), uRPFv4/v6, LPTS
- Netconf/YANG & SNMP
- Hierarchical QoS policing, WRED
- **EFD** (Early Fast Discard)
- Lawful Intercept

- IOS XR Manageability & Control Plane
  - PIE/SMU Upgrades
  - LPTS/CoPP
- Gratuitous ARP
- VRRP/HSRP

IOS XRv 9000 Operational Enhancements

Visibility & Telemetry
- Operational Data, Deep analytical hooks
- Policy-based, flexible, Push Model

Programmability
- Data accessible via published model driven interfaces
- Machine friendly
- Enables automation @ scale

Application Hosting
- Ability to run 3rd party off the shelf applications built with Linux tool chains
- Run custom applications inside an LXC container on the 64-bit Linux host

Flexible Platform and Packaging
- RPM Packages: EIGRP, MGBL, MPLS, K9SEC, LI, BGP etc.
- Automated package dependency checkers
- Automated Provisioning at Bootup
**IOS XRv 9000 Telemetry & Programmability**

- **Flexible, efficient, extensible data & transport**
- **Real-time inspection**
- **Big data aggregation and analysis**
- **Automated remediation and policy enforcement**
- **Traffic analysis, fault prediction, “gray” failure**
- **Deeper instrumentation + smarter analytics tools = Real-time action-ability**
- **Active feedback & Auto-remediation**
IOS XRv 9000 Performance

- Single Core (2015)
  - 40 Gbps
  - 8 Gbps

- Multi-Core (2016/2017)
  - 160 Gbps

- Multi-Socket (2018+)
  - 640 Gbps
  - IMIX traffic packet size with features (ACL, HQoS, Policing) enabled

Vanilla IPv4
IOS XRv 9000 Deployment
Pass-through vs Device Emulation

**Pass-through**
- Physical NIC Driver
- Hypervisor
- Hardware Platform

**Device Emulation**
- Guest Driver virtIO / E1000
- Emulated device
- Physical device+ driver
- Hypervisor
- Hardware Platform

High Performance vs Emulated
IOS XRv 9000 Deployment

Copy XRv 9000 image (.ova/.iso/.vmdk) to server
Create Disk running image
Create Virtual (Tap) interfaces
Start simulation
**IOS XRv 9000 Deployment**

**Creating TAP and Bridge**

```
cisco@linux:~$ sudo tunctl -t Tap1
Set 'Tap1' persistent and owned by uid 0

cisco@linux:~$ sudo ifconfig Tap1 up

cisco@linux:~$ sudo tunctl -t Tap2
Set 'Tap2' persistent and owned by uid 0

cisco@linux:~$ sudo ifconfig Tap2 up

<create Tap3/Tap4>
```

```
cisco@linux:~$ sudo brctl addbr vbridge1

cisco@linux:~$ sudo brctl addbr vbridge2

cisco@linux:~$ sudo brctl addif vbridge1 Tap1 eth4

cisco@linux:~$ sudo brctl addif vbridge2 Tap2 eth5

cisco@linux:~$ sudo brctl show vbridge1
```

<table>
<thead>
<tr>
<th>bridge name</th>
<th>bridge id</th>
<th>STP enabled</th>
<th>interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>vbridge1</td>
<td>8000.b6c7102ae0f6</td>
<td>no</td>
<td>Tap1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>eth4</td>
</tr>
</tbody>
</table>
IOS XRv 9000 Bring-up
Launching the IOS XRv 9000 Virtual Machine

cisco@linux:$ sudo /usr/bin/qemu-system-x86_64 \
-m 16384 \  
-smp cores=4,sockets=1 \  
-name XRv-Test-Launch \  
-drive file=./xrv9k.raw,media=disk,index=1 \  
-drive file=./xrv9k-fullk9-x.iso-6.0.0,media=cdrom,index=2 \  
-serial telnet:0.0.0.0:12345,server,nowait \  
-device e1000,netdev=mgmt-intf \  
-netdev tap,ifname=Tap1,script=no,downscript=no,id=mgmt-intf \  
-device e1000,netdev=data-intf \  
-netdev tap,ifname=Tap4,script=no,downscript=no,id=data-intf \  
-display none -enable-kvm  
-boot once=d
cisco@linux:~$ telnet 0.0.0.0 12345
[Linux-initrd @ 0x456bc000, 0x3a93367c bytes]
Starting udev
Populating dev cache
Configuring network interfaces... done.

Mon Feb  8 23:48:38 UTC 2016 (<snip>_lxc_iso.sh): Hardware profile: vpe
Mon Feb  8 23:48:38 UTC 2016 (<snip>_lxc_iso.sh): Host has 16Gb RAM / 4 vCPUs
Mon Feb  8 23:48:38 UTC 2016 (<snip>_lxc_iso.sh): Management plane: 1Gb RAM / 0 vCPUs
Mon Feb  8 23:48:38 UTC 2016 (<snip>_lxc_iso.sh): XR control plane: 7Gb RAM / 2 vCPUs
Mon Feb  8 23:48:38 UTC 2016 (<snip>_lxc_iso.sh): Data plane core assignment: 2-3
Mon Feb  8 23:48:38 UTC 2016 (<snip>_lxc_iso.sh): Control plane core assignment: 0-1
IOS Xrv 9000 Bring-up
Accessing the IOS Xrv Virtual Machine

Mon Feb 8 23:49:45 UTC 2016: Install finished on sda
Rebooting Xrv9k system after installation ...
[ 99.990922] reboot: Restarting system

# Welcome to the Cisco IOS Xrv9k platform
# Please wait for Cisco IOS XR to start.
# Copyright (c) 2014-2015 by Cisco Systems, Inc.
#
# Cisco IOS XR console will start on the 1st serial port
Cisco IOS XR aux console will start on the 2nd serial port
Cisco Calvados console will start on the 3rd serial port
Cisco Calvados aux will start on the 4th serial port

!!!!!!!!!!!!!!!!!!!! NO root-system username is configured. Need to configure root-system username.
!!!!!!!!!!!!!!!!!!!!

--- Administrative User Dialog ---
Enter root-system username:
IOS XRv 9000 Deployment
Show Commands

RP/0/RP0/CPU0:ios#show ver
Tue Feb  9 00:10:36.484 UTC
Cisco IOS XR Software, Version 6.0.0
Copyright (c) 2013-2015 by Cisco Systems, Inc.

Build Information:
Built By      : alnguyen
Built On      : Thu Dec 24 00:54:24 PST 2015
Build Host    : iox-lnx-009
Workspace     : /auto/srcarchive16/production/6.0.0/xrv9k/workspace
Version       : 6.0.0
Location      : /opt/cisco/XR/packages/
cisco IOS-XRv 9000 () processor
System uptime is 16 minutes

RP/0/RP0/CPU0:ios#
Show Commands

Node                           Type                      State       Config state
--------------------------------------------------------------------------------------
0/0/CPU0                      R-IOSXRV9000-LC-C             IOS XR RUN       NSHUT
0/RP0/CPU0                    R-IOSXRV9000-RP-C(Active)  IOS XR RUN       NSHUT

Interface                      IP-Address       Status       Protocol Vrf-Name
-------------------------------------------------------------------------------------
MgmtEth0/RP0/CPU0/0             unassigned       Shutdown     Down
GigabitEthernet0/0/0/0          unassigned       Shutdown     Down
TenGigE0/0/0/0/0                unassigned       Shutdown     Down
RP/0/RP0/CPU0/ios#              

Tap1=MgmtEth0
Tap2=Future
Tap3=Future
Tap4=Giga0/0/0/0
Tap5=TenGig0/0/0/1
IOS XRv 9000 Deployment
Containers and 3rd Party Network NameSpace

```
[sysadmin-vm:0_RP0:~]$ ssh 10.0.2.16
Last login: Tue Feb 9 01:21:24 2016 from 10.11.12.15
[host:~]$ virsh list
Id    Name                           State
----------------------------------------------------
5299  sysadmin                       running
12065 default-sdr__uvf--2            running
15153 default-sdr--1                running
[host:~]$
```

```
[sysadmin-vm:0_RP0:~]$ ssh 10.0.2.16
Last login: Tue Feb 9 01:21:24 2016 from 10.11.12.15
[host:~]$ virsh list
Id    Name                           State
----------------------------------------------------
5299  sysadmin                       running
12065 default-sdr__uvf--2            running
15153 default-sdr--1                running
[host:~]$
```

```
RP/0/RP0/CPU0:ios# show ipv4 int br
Thu Feb 11 15:55:05.581 UTC
Interface                  IP-Address      Status          Protocol Vrf
Name                       
Loopback0                  1.2.3.4          Up              Up       default
Loopback2                   110.2.2.2        Up              Up       default
Loopback3                   110.3.3.3        Up              Up       default
GigabitEthernet0/0/0/0     200.1.1.1        Up              Up       default
MgmtEth0/RP0/CPU0/0        unassigned      Shutdown        Down     default
RP/0/RP0/CPU0:ios#
```

```
[xr-vm_node0_RP0_CPU0:~]$ ip netns exec tpnns ifconfig | more
Thu Feb 11 15:55:05.581 UTC
Gi0_0_0_0 Link encap:Ethernet  HWaddr 00:50:56:b9:44:0c
inet addr:200.1.1.1  Mask:255.255.255.0
lo:0 Link encap:Local Loopback
inet addr:1.2.3.4   Mask:255.255.255.255
lo:2 Link encap:Local Loopback
inet addr:110.2.2.2  Mask:255.255.255.255
lo:3 Link encap:Local Loopback
inet addr:110.3.3.3  Mask:255.255.255.255
RP/0/RP0/CPU0:ios#
```
## IOS XRv 9000 Deployment
### Checking License Status

```
RP/0/# show license platform summary
Sat Dec 26 05:47:08.537 UTC
Current state: PRODUCTION

Collection: LAST: Sat Dec 26 05:47:03 2015
            NEXT: Sat Dec 26 06:47:03 2015
Reporting:  LAST: Sat Dec 26 05:47:03 2015
            NEXT: Sun Dec 27 05:47:03 2015

<table>
<thead>
<tr>
<th>Feature/Area</th>
<th>Entitlement</th>
<th>Last</th>
<th>Next</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Product: Right to Use</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>System</td>
<td>Feature: BGP Scale up to 4M</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
```
ISO XRv 9000 Deployment using ESXi

- ISO mage Upload
- Linux as Guest OS
- Allocated minimum 4 CPU
- Allocated minimum 8GB Mem
- Minimum 4 Network interfaces

Guest Operating System:
- Windows
- Linux
- Other

Version:
- Other 2.6.x Linux (64-bit)
IOS XRv 9000 Deployment using ESXi

Create Disk: Thin provisioning, 55GB, IDE

Create 4 x Serial Interface

1st
XR Console Port

2nd
XR AUX Port

3rd
Admin Console Port

4th
Admin AUX Port

Virtual Device Node
- SCSI (0:0)
- IDE (0:0)

Network Backing
- Server (VM listens for connection)
- Client (VM initiates connection)

Port URI: telnet://172.18.231.103:2001
IOS XRv 9000 - Deployment on ESXi
Accessing the IOS-XRv VM

Filesystem type is iso9660, using whole disk
kernel /boot/bzImage root=/dev/ram console=ttyS0 prod=1 install=/dev/sda platfo
<SNIP>
Wed Feb 17 02:13:47 UTC 2016: Copying all ISOS to repository took 68 seconds
[ 340.853307] reboot: Restarting system
Press any key to continue.
<SNIP>

#******************************************************************************#
#          Welcome to the Cisco IOS XRv9k platform                         #
# Please wait for Cisco IOS XR to start.                                   #
# Copyright (c) 2014-2015 by Cisco Systems, Inc.                          #
#******************************************************************************#
Cisco IOS XR console       will start on the 1st serial port
Cisco IOS XR aux console   will start on the 2nd serial port
Cisco Calvados console    will start on the 3rd serial port
<snip>
ios con0/RP0/CPU0 is now available
Press RETURN to get started.
!!!!!!!!!!!!!!!!!!!!!! NO root-system username is configured. Need to configure root-
!!!!!!!!!!!!!!!!!!!!!!Configuration lock is held by another agent. Please wait. 

--- Administrative User Dialog ---
Enter root-system username:

Telnet to the Serial Port
`telnet <esxi_host_ip> <port_number>`

Will go through baking process on first boot up &reload
Only happens once, during the first bootup

Create Username and Password
IOS XRv 9000 Deployment using Openstack

Create Flavor → Create Network → Create Image → Disk → RAM → Select Flavor → Launch Instance
IOS XRv 9000 Deployment using Openstack

Create Image

Assign Network

Boot Source
IOS XRv 9000 Deployment on AWS

The Cisco XRv9000 is a 64Bit Carrier Grade Virtual Router providing both virtual provider edge (vPE) and route reflector (vRR) capabilities. The AMI image runs Cisco's IOS-XR technology enabling customers to deploy the same world class networking services that they are used to in their own networks but inside AWS. It includes the following functionality: IOS-XR Base Tech Package: BGP, OSPF, ISIS, IPv4, IPv6, GRE (IPv4 transport with IPv4 or IPv6 payload), uRPF, NTP, QoS ((Policing/Marking/H-QoS/Egress-TM), Hierarchical Policers (conform aware), 802.1Q VLAN, ACL, AAA, RADIUS, TACACS+, IOS-XR CLI, SSH, SNMP, and FFM.
IOS XRv Troubleshooting
IOS XRv 9000 Troubleshooting

Hierarchy

- Host Hardware
- Host OS
- HyperVisor
- IOS XRv 9000
  - Fwering
  - Routing
  - Mgmt
  - Admin
- XR Level
- VNF Level
- Hypervisor Level
- Host OS Level
- Hardware Level
## Regular XR Troubleshooting Techniques

### DPA / DPC Communication and Packet Stats

Non-Uniform-Memory-Access (NUMA) Aware

Hyper-threading (HT) Agnostic

```
RP/0/RP0/CPU0:SS_Node1#show controllers dpa statistics global

<table>
<thead>
<tr>
<th>Index</th>
<th>Punt</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1575</td>
<td>ARP</td>
<td>10</td>
</tr>
<tr>
<td>1677</td>
<td>IFIB</td>
<td>104034</td>
</tr>
<tr>
<td>1698</td>
<td>IPv4 BFD</td>
<td>1404379</td>
</tr>
<tr>
<td>1722</td>
<td>IPv4 incomplete TX adjacency</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Index</th>
<th>Inject</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>268</td>
<td>IPv4 from fabric multicast</td>
<td>103716</td>
</tr>
<tr>
<td>270</td>
<td>IPv4 from fabric next-hop</td>
<td>330</td>
</tr>
<tr>
<td>275</td>
<td>Inject to fabric</td>
<td>104047</td>
</tr>
<tr>
<td>276</td>
<td>Inject to port</td>
<td>1510764</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Index</th>
<th>Drop</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>85</td>
<td>IPv4 disabled in uIDB</td>
<td>3888</td>
</tr>
<tr>
<td>113</td>
<td>IPv6 disabled in uIDB</td>
<td>60</td>
</tr>
<tr>
<td>236</td>
<td>Preroute PIT lookup missed</td>
<td>1</td>
</tr>
</tbody>
</table>
```
IOS XRv 9000 Troubleshooting

Hierarchy

- XR Level
  - Intel Virtualization turned on in BIOS
  - Power optimization turn off

- VNF Level
  - Dedicated pinned CPU (vCPU = pCPU) for best performance
  - Non-Uniform-Memory-Access (NUMA)
    - use local memory same node as dedicated CPU, not foreign memory

- Hypervisor Level

- Host OS Level

- Hardware Level
  - HyperTreading (HT) agnostic
  - Turn Off Power Optimization
server!~$ sudo netstat -pln | grep 12346
tcp 0 0 0.0.0.0:12346 0.0.0.0:* LISTEN 16135/qemu-system-
x

server!~$ numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7 16 17 18 19 20 21 22 23
node 0 size: 257762 MB
node 0 free: 194589 MB
node 1 cpus: 8 9 10 11 12 13 14 15 24 25 26 27 28 29 30 31
node 1 size: 258045 MB
node 1 free: 247971 MB

IOS XRv 9000 Performance Troubleshooting
IOS XRv 9000 Performance Troubleshooting

```
top - 12:58:16 up 64 days, 13:03,  4 users,  load average: 2.63, 2.58, 2.24
Tasks: 443 total,   2 running, 441 sleeping,   0 stopped,   0 zombie
%Cpu0 :  1.7 us,  0.3 sy,  0.7 ni, 97.2 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu1 : 99.7 us,  0.3 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu2 :  1.0 us,  0.0 sy,  0.0 ni, 99.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu3 :  1.0 us,  0.0 sy,  0.0 ni, 99.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu4 :  1.7 us,  0.3 sy,  0.0 ni, 97.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu5 :  7.2 us,  3.8 sy,  0.0 ni, 89.1 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu6 :  1.3 us,  0.0 sy,  0.0 ni, 98.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu7 :  1.3 us,  0.0 sy,  0.0 ni, 98.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu8 : 100.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu9 :  0.0 us,  0.3 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu10:  0.7 us,  0.3 sy,  0.0 ni, 99.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu11:  1.3 us,  0.0 sy,  0.0 ni, 98.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu12:  3.0 us,  1.3 sy,  0.0 ni, 95.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu13:  1.0 us,  0.0 sy,  0.0 ni, 99.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st

cisco@uLinux-4:~$ sudo numastat -p 18135

Per-node process memory usage (in MBs) for PID 18135 (qemu-system-x86)

<table>
<thead>
<tr>
<th></th>
<th>Node 0</th>
<th>Node 1</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Huge</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Heap</td>
<td>0.71</td>
<td>1.37</td>
<td>2.08</td>
</tr>
<tr>
<td>Stack</td>
<td>0.07</td>
<td>0.03</td>
<td>0.10</td>
</tr>
<tr>
<td>Private</td>
<td>9735.30</td>
<td>372.94</td>
<td>10108.23</td>
</tr>
<tr>
<td></td>
<td>9736.07</td>
<td>374.34</td>
<td>10110.41</td>
</tr>
</tbody>
</table>
```
server!~$ sudo taskset -pc 1 18135
pid 18135's current affinity list: 0-31
pid 18135's new affinity list: 1

server!~$ sudo taskset -pc 2 18135
pid 18135's current affinity list: 1
pid 18135's new affinity list: 2
IOS XRv 9000 Troubleshooting

Hypervisor Common mistakes

• Multiple XRv 9000 using same disk image
  – Each instance needs a separate disk

• Multiple XRv using same console

```
$ qemu-system-x86_64 -serial telnet::2345,server,nowait <...snip...>

inet_listen_opts: bind(ipv4,0.0.0.0,2345): Address already in use
inet_listen_opts: FAILED
chardev: opening backend "socket" failed: Address already in use
qemu: could not open serial device 'telnet:0.0.0.0:2345,server,nowait': Address already in use
```
IOS XRv 9000 Troubleshooting

Host Commands

Process & CPU Status

top - 09:26:59 up 13 days, 58 min, 2 users, load average: 0.49, 0.47, 0.50
Tasks: 325 total, 2 running, 322 sleeping, 0 stopped, 1 zombie
Cpu(s): 1.6%us, 0.4%sy, 0.1%ni, 98.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 181507016k total, 8715152k used, 172791864k free, 146552k buffers
Swap: 134143996k total, 0k used, 134143996k free, 4476556k cached

PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
5477 cisco     20   0 50460  26m 3808 R  39  0.0   4843:57 Xtightvnc
31187 libvirt- 20   0 5972m 1.9g 7640 S    9  1.1 377:28.36 qemu-system-x86
34605 libvirt- 20   0 5972m 1.8g 7628 S    7  1.0 348:28.40 qemu-system-x86
24982 cisco     30  10 32068 1844 1488 S    4  0.0   0:05.99 fuzzyflakes
IOS XRv 9000 Troubleshooting

Host OS Common Issues

• Missing qemu/KVM package
  – apt-get install qemu-kvm;

• Not able to create virtual interface:
  – apt-get install uml-utilities (required for tap interface for bridge to physical)

• Virtualization not enabled:
  – Ensure VT-x flag (or equivalent) is exposed to operating system
    • egrep -c '(vmx|svm)' /proc/cpuinfo

• Ensure that user is added to ‘kvm’ group
  – sudo addgroup `id -un` kvm

```bash
server!~$ sudo kvm-ok
INFO: /dev/kvm exists
KVM acceleration can be used
```
Service Orchestration for NFV
Service Orchestration for IOS XRv 9000

Network Function Virtualization

Software Defined Networking

Service Orchestration

Cloud VM Orchestration
IOS XR 9000 as VNF – Flexible Choices

Right Sizing Your Deployment

Choose your Service
- Core / Transport
- Peering
- DCI
- PE
- Subscriber Services
- Virtual PE (vPE)
- Virtual RR (vRR)

Size up your Data Plane
- Multichassis NCS 6000
- Multichassis CRS-X
- NCS 6000
- NCS 5500
- ASR 9000 Tomahawk
- CRS-X
- NCS5500/CRS
- ASR 9000 Typhoon
- IOS-XRv 9K

Adapt your Control Plane
- Control Plane
  - Choose between On-box, Hybrid or Pure Virtual CP Based on Use Case
- Today's IOS-XR on box Control Plane
- Virtualized CP or Expansion CP from Physical System

Data Plane
- Low
- High

Control Plane
- Low
- High
NFV with IOS XR
Completing Portfolio

- CRS Portfolio
- Edge Routing ASR 9000 Portfolio
- NCS 5500
- NCS 6000 Single & Multi Chassis
- NFV Virtual Router IOS-XRv 9000
Putting it all together…

- SMU-ability
- Opex Saving
- Low Capex
- Carrier Class
- High Availability
- Scalable
- Multi-threaded
- Flexible
- Elastic

IOS XRv 9000
Virtual Router
Real Performance
Re-Cap

• Role of NFV in Network
• IOS XRv 9000
• Benefits & Use Cases
• Deployment & Troubleshooting
• Service Orchestration for NFV
• Summary
By the same author…

ISBN: 978-0134463056
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#BRKSPG-2724
• 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.ciscobackup.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
You’re

Cisco Live!