排除C8000v路由器的性能问题

目录

简介

使用的组件

一般故障排除

- 超支
- 功能丢弃
- 尾滴

虚拟机监控程序

- VMware ESXi
- AWS
  - 多TX队列
  - 超过指标
- Microsoft Azure
  - 加速网络
  - Azure和分段
  - Microsoft Azure支持的实例类型

其它资源

简介

本文档介绍如何解决C8000v企业路由器在公共云和ESXi方案中的性能问题。

使用的组件

本文档中的信息基于以下硬件和软件组件：

- 运行17.12版本的C8000v
- ESXI版本7.0 U3

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

一般故障排除

虽然您可以在不同的环境中托管C8000v，但无论托管C8000v的位置如何，您仍然可以完成一些相同的故障排除步骤。让我们从基本知识开始。您需要确保的第一件事是设备是否达到其容量限制。为此，您可以首先检查以下两个输出：

1.show platform hardware qfp active datapath util summary — 此命令为您提供C8000v从每个端口接收和传输的输入/输出的完整信息。您必须将注意力集中在处理负载百分比上。如果您处于达到100%的场景中，则意味着您将达到容量限制

```
------------------ show platform hardware qfp active datapath utilization summary  ------------------

  CPP 0:                     5 secs        1 min        5 min       60 min
Input:     Total (pps)        93119        92938        65941        65131
                 (bps)    997875976   1000204000    708234904    699462016
Output:    Total (pps)        93119        92949        65944        65131
                 (bps)   1052264704   1054733128    746744264    737395744
Processing: Load (pct)           14           14           10           10
```

2.show platform hardware qfp active datapath infrastructure sw-cio — 将此命令视为以上命令更深入的版本。它提供有关单个内核（包括IO和加密内核，它们不是QFP利用率数字的一部分）的更详细的信息。当您想要查看特定数据平面核心是否导致瓶颈时，它非常有用。



提示：使用此命令时需要非常重要的详细信息，请始终运行两次。当计算执行命令之间的核心利用率百分比时。

```
------------------ show platform hardware qfp active datapath infrastructure sw-cio  ------------------

Credits Usage:

  ID      Port  Wght  Global WRKR0  WRKR1  WRKR2  WRKR3  WRKR4  WRKR5  WRKR6  WRKR7  WRKR8  WRKR9  WRKR
   1      rcl0    16:    492      0      0      0      0      0      0      0      0      0
   1      rcl0    32:    496      0      0      0      0      0      0      0      0      0
   2       ipc     1:    489      0      0      0      0      0      0      0      0      0
   3 vxe_punti     4:    490      0      0      0      0      0      0      0      0      0
   4       Gi1     4:   1999      0      0      0      0      0      0      0      0      0
   5       Gi2     4:   1991      0      0      0      0      0      0      0      0      0
   6       Gi3     4:   1991      0      0      0      0      0      0      0      0      0
   7       Gi4     4:   1993      0      0      0      0      0      0      0      0      0
   8       Gi5     4:   2009      0      0      0      0      0      0      0      0      0
   9       Gi6     4:   2015      0      0      0      0      0      0      0      0      0
  10       Gi7     4:   2002      0      0      0      0      0      0      0      0      0
  11      vpg0   400:    490      0      0      0      0      0      0      0      0      0

Core Utilization over preceding 107352.2729 seconds
---------------------------------------------------
      ID:       0       1       2       3       4       5       6       7       8       9      10
    % PP:    2.98    2.01    1.81    1.67    1.60    1.53    1.35    1.30    1.25    1.19    2.19
    % RX:    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
    % TM:    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
```

```
% IDLE:   97.02   97.99   98.19   98.33   98.40   98.47   98.65   98.70   98.75   98.81   97.81   98.
```

现在，您已确定您是否达到平台限制。下一步是检查丢包。这些与性能问题密切相关。根据丢弃发生的位置，可以考虑三种类型的丢弃。

- 超支：这种类型的数据包丢弃发生在Rx端。发生这种情况是因为已超出一个或多个内核的处理能力。
- 功能丢弃：此类数据包丢弃发生在PPE中。它们与路由器中的功能（例如ACL或QoS）相关。
- 尾部丢弃：这种类型的数据包丢弃发生在Tx端。它们是由于Tx缓冲区的拥塞引起的。

要确定您遇到了哪些丢弃，可以使用以下输出：

- show platform hardware qfp active drop state clear
- show interface
- show policy map interface

您将验证如何确定您面临哪些丢包以及如何缓解这些丢包。然而，本文的最大关注点将是称为Taildrops的丢包，因为在虚拟路由器中排除故障非常棘手。

超支

当网络接口接收数据包的速度超过其处理或存储数据包的速度时，Cisco IOS XE会发生溢出丢弃。具体而言，接口（FIFO队列）的内部缓冲区已满，因为传入数据速率超过了硬件处理它的能力。因此，新的传入数据包无法存储并被丢弃，这会增加超限计数器。这基本上是由于接口暂时不堪重负而导致的数据包丢失。

这种类型的数据包丢弃发生在Rx端。发生这种情况是因为已超出一个或多个内核的处理能力，并且Rx线程无法将传入数据包分发到相关PP线程，并且入口缓冲区已满。打个简单的比方，您可以将其视为结账柜台的队列，由于数据包到达的速度快于出纳员（接口硬件）可以提供的速度，因此队列过满。当队列已满时，新客户必须离开而得不到服务，这是超支的丢包。

尽管本节中提到了硬件，但C8000v是基于软件的路由器。在这种情况下，超支可能是由以下原因造成的：

- 高数据平面利用率：如果数据平面利用率很高，则数据包轮询速度不够快，导致超载。例如，“大象流”（大型、连续的数据流）的存在可能会使处理资源饱和，并导致接口上的超载。
- 设备模板不正确：使用不适当的设备模板可能会导致低效的缓冲区管理和溢出。可以使用show platform software cpu alloc命令检查此情况，也可以使用platform resource <template>命令进行更改。

每个接口都分配了一个有限的信用池，此机制可防止接口繁忙和系统资源过载。每次新数据包到达数据平面时，都需要信用额度。当数据包处理完成时，将返回积分，以便Rx线程可以再次使用它。如果没有用于该接口的可用信用，则数据包需要在接口Rx环中等待。通常，您预计与丢包相关的性能限制将为Rx溢出，因为已超过一个或多个内核的处理能力。

要识别溢出，通常需要检查接口统计信息中的输入错误，特别是溢出计数器：

- 使用命令show platform hardware qfp active datapath infrastructure sw-cio确定核心利用率，以及是否已超过特定接口的信用数量。
- 使用命令show interface <interface-name>在输出中查找溢出计数。

超限显示为输入错误的一部分，例如：

```
Gig2 is up, line protocol is up
241302424557 packets input, 168997587698686 bytes, 0 no buffer
20150 input errors, 0 CRC, 0 frame, 20150 overrun, 0 ignored <<<<<<<<<<<
```

让我们假设Gig2观察由超限引起的性能问题的情况。要确定与此接口关联的工作线程，您可以使用以下命令：

```
show platform hardware qfp active datapath infra binding
Port Instance Bindings:

  ID       Port                 IOS Port WRKR 2
   1       rcl0                     rcl0  Rx+Tx
   2        ipc                      ipc  Rx+Tx
   3  vxe_punti              vxe_puntif  Rx+Tx
   4        Gi1         GigabitEthernet1  Rx+Tx
   5        Gi2         GigabitEthernet2  Rx+Tx >> in this case, WRKR2 is the thread responsible for bo
```

然后，您可以分析负责此接口的Rx流量的特定线程的利用率及其信用数。

在Gig2观察因超限引起的性能问题的情况下，可以预期PP#2会持续完全利用（空闲= 0%），而接口Gig2的信用点为低/零：

```
show platform hardware qfp active datapath infrastructure sw-cio
Credits Usage:

  ID      Port  Wght  Global WRKR0  WRKR1  WRKR2  Total
   1      rcl0    16:    487     0      0     25    512
```

```
   1      rcl0    32:    496     0      0     16    512
   2       ipc     1:    490     0      0     21    511
   3 vxe_punti     4:    459     0      0     53    512
   4       Gi1     4:    477     0      0     35    512
   5       Gi2     4:    474     0      0     38    512    <<< low/zero credits for interface Gig2:

Core Utilization over preceding 1.0047 seconds
-----------------------------------------------
      ID:       0       1       2
    % PP:    0.77    0.00    0.00
    % RX:    0.00    0.00    0.44
    % TM:    0.00    0.00    5.63
  % IDLE:   99.23   99.72   93.93    <<< the core ID relevant in this case would be PP#2
```

功能丢弃

数据包由任何可用的数据平面线程处理，并且严格基于QFP核心的可用性通过软件Rx功能(x86) — 基于负载的分配(LBD)进行分配。到达PPE的数据包可以因特定QFP丢弃原因而被丢弃，可以使用以下输出进行检查：

```
------------------ show platform hardware qfp active statistics drop detail ------------------

Last clearing of QFP drops statistics : never

-------------------------------------------------------------------------------
   ID  Global Drop Stats                          Packets                Octets
-------------------------------------------------------------------------------
  319  BFDoffload                                     403                 31434
  139  Disabled                                       105                  7487
   61  Icmp                                           135                  5994
   94  Ipv4NoAdj                                        1                   193
   33  Ipv6NoRoute                                   2426                135856
  215  UnconfiguredIpv4Fia                        1937573             353562196
  216  UnconfiguredIpv6Fia                        8046173            1057866418


------------------ show platform hardware qfp active interface all statistics drop_summary ------------

-----------------------------------------------------------------
Drop Stats Summary:
note: 1) these drop stats are only updated when PAL
         reads the interface stats.
      2) the interface stats include the subinterface

Interface                                   Rx Pkts            Tx Pkts
-------------------------------------------------------------------------
GigabitEthernet1                            9980371                  0
GigabitEthernet2                               4012                  0
```

下降的原因多种多样，通常不言自明。为了进一步研究，可以使用数据包跟踪。

尾滴

如前所述，当设备尝试传输数据包但传输缓冲区已满时，就会发生尾部丢弃。

在此小节中，您将查看在遇到此类情况时可以检查的输出。您在这些文档中看到的价值意味着什么？您可以采取哪些措施缓解此问题？

首先，你需要知道如何识别它们。其中一种方法就是简单地查看show interface。请留意不断增加的输出滴数：

```
GigabitEthernet2 is up, line protocol is up
  Hardware is vNIC, address is 0050.56ad.c777 (bia 0050.56ad.c777)
  Description: Connected-To-ASR Cloud Gateway
  Internet address is 10.6.255.81/29
  MTU 1500 bytes, BW 10000000 Kbit/sec, DLY 10 usec,
     reliability 255/255, txload 2/255, rxload 3/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full Duplex, 10000Mbps, link type is force-up, media type is Virtual
  output flow-control is unsupported, input flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters 03:16:21
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 7982350 >>> Output drops.
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 150449000 bits/sec, 20461 packets/sec
  5 minute output rate 89116000 bits/sec, 18976 packets/sec
```

此命令对于了解您是否遇到拥塞特别有用：

- show platform hardware qfp active datapath infrastructure - HQF表示“Hierarchical Queueing Framework”。此功能支持使用模块化QoS命令行界面(MQC)进行不同级别（物理、逻辑和类）的服务质量(QoS)管理。 它显示当前的RX和TX成本。 当TX队列已满时，如输出所示（1959年已满）

```
pmd b1689fc0 device Gi1
RX: pkts 5663120 bytes 1621226335 return 0 badlen 0
Out-of-credits: Hi 0 Lo 0
pkts/burst 1 cycl/pkt 1565 ext_cycl/pkt 1173
Total ring read 12112962299, empty 12107695202
```

```
TX: pkts 8047873582 bytes 11241140363740
pri-0: pkts 8047873582 bytes 11241140363740
pkts/send 3
Total: pkts/send 3 cycl/pkt 452
send 2013612969 sendnow 1810842
forced 2013274797 poll 724781 thd_poll 0
blocked 2197451 retries 7401 mbuf alloc err 0
TX Queue 0: full 1959 current index 0 hiwater 224 >>>
```

输出表明底层硬件没有跟上数据包的发送速度。要调试底层接口，可能需要查看C8000v的外部，以及C8000v运行所在的底层环境，以查看底层物理接口上是否报告了任何其他错误。

要检查环境，在检查C8000v路由器的哪个虚拟机监控程序运行之前，您可以执行一个步骤。这是为了检查命令show controller输出。然而，你可能会迷失在每一个柜台的意义或去哪里看东西上。

首先，查看此输出时，您需要记住的一个重要详细信息是，信息主要来自vNIC本身。每个NIC驱动程序都有一组他们使用的特定计数器，这些计数器可能会因驱动程序而异。不同的虚拟机监控程序对显示的内容也有一定的影响。某些计数器（例如mbuf计数器）是DPDK驱动程序的统计信息。这些DPDK驱动程序因不同的DPDK驱动程序而异。实际计数通常由虚拟化层的虚拟机监控程序完成。

```
GigabitEthernet2 - Gi2 is mapped to UIO on VXE
rx_good_packets 1590
tx_good_packets 1402515
rx_good_bytes 202860
tx_good_bytes 1857203911
rx_missed_errors 0
rx_errors 0
tx_errors 0
rx_mbuf_allocation_errors 0
rx_q0_packets 1590
rx_q0_bytes 202860
rx_q0_errors 0
tx_q0_packets 1402515
tx_q0_bytes 1857203911
rx_q0_drop_total 0
rx_q0_drop_err 0
rx_q0_drop_fcs 0
rx_q0_rx_buf_alloc_failure 0
tx_q0_drop_total 976999540797
tx_q0_drop_too_many_segs 0
tx_q0_drop_tso 0
tx_q0_tx_ring_full 30901211518
```

请点击此处了解如何解释和阅读这些计数器：

1. 如果看到subX，则表示它是子接口 — 主接口的逻辑划分。sub0通常是主/默认的。当涉及多个VLAN时，通常使用这些方法。
2. 然后，您有“rx =接收”和“tx =发送”。

3. 最后，q0是指该接口使用的第一个/默认队列

虽然没有对每个计数器进行说明，但文章介绍了其中一些计数器，它们可能与您的故障排除有关：

- 当NIC缓冲区(Rx FIFO)过满时，出现“RX_MISSED_ERRORS：”。这种情况会导致丢包和延迟增加。一个可能的解决方案是增加网卡缓冲区（在我们的情况下是不可能的）或更改网卡驱动程序。
- "tx_q0_drop_total"和"tx_q0_tx_ring_full":这可以告诉您主机正在丢弃数据包，而且C8000v在C8000v中遇到尾部丢弃，因为主机正在对C8000v进行反压

在以上输出中，我们未看到任何“rx_missed_errors”。但是，由于我们关注的是尾部丢弃，因此我们可以看到“tx_q0_drop_total”和“tx_q0_tx_ring_full”。 这样，我们可以断定，确实存在由主机底层硬件导致的拥塞。

如前所述，每个虚拟机监控程序都会对显示的内容产生某种影响。下一节将重点介绍这一点，因为它将讨论可以托管C8000v的不同虚拟机监控程序之间的差异。您还可以找到不同的建议，尝试缓解每个建议中的此类问题。

虚拟机监控程序

hypervisor是一个软件层，它通过管理和分配硬件资源（如CPU、内存和存储）到每个VM，允许多个操作系统（称为虚拟机或虚拟机）在单个物理硬件主机上运行。它可确保这些虚拟机独立运行，互不干扰。

在Cisco Catalyst 8000V(C8000v)中，虚拟机监控程序是托管C8000v虚拟机的平台。您如何确定托管C8000v的虚拟机监控程序？有一个相当有用的输出为我们提供了这些信息。此外，您还可以检查我们的虚拟路由器可以访问哪些类型的资源：

```
C8000v#show platform software system all
Processor Details
=================
Number of Processors : 8
Processor : 1 - 8
vendor_id : GenuineIntel
cpu MHz : 2593.906
cache size : 36608 KB
Crypto Supported : Yes
model name : Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz

Memory Details
==============
Physical Memory : 32817356KB

VNIC Details
============
```

```
Name Mac Address Driver Name Status Platform MTU
GigabitEthernet1 0022.480d.7a05 net_netvsc UP 1500
GigabitEthernet2 6045.bd69.83a0 net_netvsc UP 1500
GigabitEthernet3 6045.bd69.8042 net_netvsc UP 1500

Hypervisor Details
==================
Hypervisor: AZURE
Manufacturer: Microsoft Corporation
Product Name: Virtual Machine
Serial Number: 0000-0002-0201-5310-5478-4052-71
UUID: 8b06091c-f1d3-974c-85a5-a78dfb551bf2
Image Variant: None
```

VMware ESXi

ESXi是由VMware开发的第1类虚拟机监控程序，它直接安装在物理服务器上以实现虚拟化。它通过提取硬件资源并将其分配到每个虚拟机，使多个虚拟机(VM)可以在单个物理服务器上运行。C8000v路由器就是其中一个VM。

您可以从出现拥塞的常见场景开始。这可以通过检查tx_q0_tx_ring_full计数器来确认：

示例：

```
------------------ show platform software vnic-if interface-mapping ------------------

------------------------------------------------------------
Interface Name Driver Name Mac Addr
------------------------------------------------------------
GigabitEthernet3 net_vmxnet3 <-- 0050.5606.2239
GigabitEthernet2 net_vmxnet3 0050.5606.2238
GigabitEthernet1 net_vmxnet3 0050.5606.2237
------------------------------------------------------------

GigabitEthernet3 - Gi3 is mapped to UIO on VXE
rx_good_packets 99850846
tx_good_packets 24276286
rx_good_bytes 78571263015
tx_good_bytes 14353154897
rx_missed_errors 0
rx_errors 0
tx_errors 0
rx_mbuf_allocation_errors 0
rx_q0packets 99850846
rx_q0bytes 78571263015
rx_q0errors 0
tx_q0packets 24276286
tx_q0bytes 14353154897
rx_q0_drop_total 0
rx_q0_drop_err 0
```

```
rx_q0_drop_fcs 0
rx_q0_rx_buf_alloc_failure 0
tx_q0_drop_total 160945155
tx_q0_drop_too_many_segs 0
tx_q0_drop_tso 0
tx_q0_tx_ring_full 5283588 <------
```

当C8000V尝试通过VMXNET3接口发送数据包时，会发生此拥塞。但是，缓冲区环已满是数据包，导致延迟或丢弃。

在这些情况下，这些丢弃发生在前面提到的虚拟机监控程序端。如果满足所有建议，建议与VMware支持人员联系，以了解网卡上发生的情况。

下面是有关如何提高性能的一些建议：

- 使用专用vSwitch和上行链路以获得最佳性能
- 通过将C8000V分配给由自己的物理上行链路支持的专用vSwitch，我们可以将其流量与噪声邻居隔离，并避免共享资源瓶颈。

有一些命令值得从ESXi端查看。例如，要检查来自ESXi接口的数据包丢失，可以执行以下操作：

1. 启用 SSH.
2. 使用SSH连接到ESXi。
3. 运行esxtop。
4. 键入n。

如果虚拟机的网络驱动程序耗尽Rx缓冲区内存，esxtop命令可以显示虚拟交换机上丢弃的数据包。即使esxtop显示数据包在虚拟交换机处被丢弃，它们实际上也会在虚拟交换机和访客操作系统驱动程序之间被丢弃。

搜索%DRPTX和%DRPRX下丢弃的所有数据包：

```
12:34:43pm up 73 days 16:05, 907 worlds, 9 VMs, 53 vCPUs; CPU load average: 0.42, 0.42, 0.42

PORT-ID     USED-BY                    TEAM-PNIC   DNAME              PKTTX/s   MbTX/s   PSZTX    PKTRX/s
67108870    Management                  n/a         vSwitch-to-9200      0.00     0.00     0.00      0.00
67108872    Shadow of vmnic1           n/a         vSwitch-to-9200      0.00     0.00     0.00      0.00
67108876    vmk1                       vmnic1      vSwitch-to-9200      0.00     0.00     0.00      0.00
67108890    2101719:c8kv-gw-mgmt       vmnic1      vSwitch-to-9200   76724.83   792.35  1353.00  16180.39
100663305   Management                  n/a         vSwitch-to-Cisc      0.00     0.00     0.00      0.00
100663307   Shadow of vmnic0           n/a         vSwitch-to-Cisc      0.00     0.00     0.00      0.00
100663309   vmk0                       vmnic0      vSwitch-to-Cisc      3.64     0.01   280.00      3.29
```

```
100663310   2100707:gsoaresc-On_Prem  vmnic0      vSwitch-to-Cisc      0.00     0.00     0.00     2.43
100663311   2100993:cats-vmanage      void        vSwitch-to-Cisc      0.00     0.00     0.00     0.00
100663312   2100993:cats-vmanage      void        vSwitch-to-Cisc      0.00     0.00     0.00     0.00
100663313   2100993:cats-vmanage      vmnic0      vSwitch-to-Cisc      5.38     0.01   212.00     9.71
100663314   2101341:cats-vsmart       void        vSwitch-to-Cisc      0.00     0.00     0.00     0.00
100663315   2101341:cats-vsmart       vmnic0      vSwitch-to-Cisc      2.60     0.00   164.00     6.94
100663316   2101522:cats-vbond        vmnic0      vSwitch-to-Cisc      0.00     0.00     0.00     0.00
100663317   2101522:cats-vbond        vmnic0      vSwitch-to-Cisc      0.00     0.00     0.00     0.00
100663318   2101522:cats-vbond        vmnic0      vSwitch-to-Cisc      4.33     0.01   174.00     7.80
100663319   2101522:cats-vbond        vmnic0      vSwitch-to-Cisc      0.00     0.00     0.00     4.16
100663320   2101547:gdk-backup        vmnic0      vSwitch-to-Cisc      0.00     0.00     0.00     3.12
100663321   2101703:sevvy             vmnic0      vSwitch-to-Cisc      0.00     0.00     0.00     3.12
100663323   2101719:c8kv-gw-mgmt      vmnic0      vSwitch-to-Cisc  16180.91     9.09    73.00  76755.87
100663324   2137274:telemetry-server  vmnic0      vSwitch-to-Cisc      0.00     0.00     0.00     3.12
100663335   2396721:netlab            vmnic0      vSwitch-to-Cisc      0.00     0.00     0.00     3.12
2214592519  vmnic1                    -           vSwitch-to-9200  76727.26   792.38  1353.00  16182.64
2248146954  vmnic0                    -           vSwitch-to-Cisc  16189.05     9.32    75.00  76736.97
```

此命令列出主机上配置的所有NIC:

```
esxcli network nic list
Name PCI Device Driver Admin Status Link Status Speed Duplex MAC Address MTU Description
------ ------------ ------ ------------ ----------- ----- ------ ----------------- ---- -----------
vmnic0 0000:01:00.0 igbn Up Up 1000 Full fc:99:47:49:c5:0a 1500 Intel(R) I350 Gigabit Network Connectio
vmnic1 0000:01:00.1 igbn Up Up 1000 Full fc:99:47:49:c5:0b 1500 Intel(R) I350 Gigabit Network Connectio
vmnic2 0000:03:00.0 ixgben Up Up 1000 Full a0:36:9f:1c:1f:cc 1500 Intel(R) Ethernet Controller 10 Gigab
vmnic3 0000:03:00.1 ixgben Up Up 1000 Full a0:36:9f:1c:1f:ce 1500 Intel(R) Ethernet Controller 10 Gigab
```

还有一个有用的命令可用于检查分配给特定VM的vNIC的状态。

```
esxcli network vm list
World ID  Name                Num Ports  Networks
--------  ------------------  ---------  ----------------------------------------
2137274   telemetry-server    1          Cisco Backbone 10.50.25.0/24
2101703   sevvy               1          Cisco Backbone 10.50.25.0/24
2396721   netlab              1          Cisco Backbone 10.50.25.0/24
2101547   gdk-backup          1          Cisco Backbone 10.50.25.0/24
2101522   cats-vbond          4          VPN0, VPN0, VPN0, VPN0
2101719   c8kv-gw-mgmt        2          c8kv-to-9200l, c8kv-to-cisco
2100707   gsoaresc-On_Prem    1          Cisco Backbone 10.50.25.0/24
2100993   cats-vmanage        3          VPN0, VPN0, VPN0
2101341   cats-vsmart         2          VPN0, VPN0
```

查看c8kv-gw-mgmt（C8000v虚拟机），分配了2个网络：

- c8kv-to-9200l

- c8kv-to-cisco

您可以使用全局ID查找有关此VM的详细信息：

```
[root@localhost:~] esxcli network vm port list -w 2101719
Port ID:           67108890
vSwitch:           vSwitch-to-9200L
Portgroup:         c8kv-to-9200l
DVPort ID:
MAC Address:       00:0c:29:31:a6:b6
IP Address:        0.0.0.0
Team Uplink:       vmnic1
Uplink Port ID:    2214592519
Active Filters:

Port ID:           100663323
vSwitch:           vSwitch-to-Cisco
Portgroup:         c8kv-to-cisco
DVPort ID:
MAC Address:       00:0c:29:31:a6:ac
IP Address:        0.0.0.0
Team Uplink:       vmnic0    <----
Uplink Port ID:    2248146954
Active Filters:
[root@localhost:~]
```

获得此信息后，您可以确定vSwitch被分配到哪个网络。

要检查分配给vSwitch的物理NIC的一些流量统计信息，可使用以下命令：

```
# esxcli network nic stats get -n <vmnic>
```

此命令显示信息，例如接收的数据包、接收的字节数、丢弃的数据包和接收的错误。这有助于确定网卡上是否发生丢包。

```
[root@localhost:~] esxcli network nic stats get -n vmnic0
NIC statistics for vmnic0
Packets received: 266984237
Packets sent: 123640666
Bytes received: 166544114308
Bytes sent: 30940114661
Receive packets dropped: 0
Transmit packets dropped: 0
Multicast packets received: 16773454
Broadcast packets received: 36251726
```

```
Multicast packets sent: 221108
Broadcast packets sent: 1947649
Total receive errors: 0
Receive length errors: 0
Receive over errors: 0
Receive CRC errors: 0
Receive frame errors: 0
Receive FIFO errors: 0
Receive missed errors: 0
Total transmit errors: 0
Transmit aborted errors: 0
Transmit carrier errors: 0
Transmit FIFO errors: 0
Transmit heartbeat errors: 0
Transmit window errors: 0
```

可以通过修改主机和虚拟机上的设置来验证一些可以提高ESXi环境中运行的Cisco Catalyst 8000V性能的配置：

- 设置虚拟硬件：CPU预留设置为最大值。
- 在虚拟硬件中保留所有访客内存：内存。
- 从虚拟硬件中选择VMware Paravirtual:SCSI控制器。
- 从虚拟硬件：网络适配器：适配器类型选项，为支持的NIC选择SR-IOV
- 将General Guest OS Version > VM Options选项设置为Other 3.x or later Linux（64位）。
- 将Advanced Latency Sensitivity下的VM Options选项设置为High。
- 在VM Options > Advanced Edit Configuration下，将“numa.nodeAffinity”添加到与SRIOV NIC相同的NUMA节点
- 启用虚拟机监控程序性能设置。
- 通过在支持的物理NIC上启用SR-IOV来限制vSwitch的开销。
- 将VM的vCPU配置为与物理NIC在同一NUMA节点上运行。
- 将VM Latency Sensitivity（VM延迟灵敏度）设置为High（高）。

AWS

C8000v通过在Amazon虚拟私有云(VPC)中作为Amazon Machine Image(AMI)启动，支持在AWS上部署，允许用户为他们的网络资源调配AWS云的逻辑隔离部分。

多TX队列

在AWS上运行的C8000v中，一项关键功能是使用多TX队列（多TXQ）。 这些队列有助于减少内部处理开销并提高可扩展性。使用多个队列可以更快更简单地为正确的虚拟CPU(vCPU)分配传入和传出数据包。

与某些每个vCPU分配RX/TX队列的系统不同，在C8000v中，这些队列按接口分配。RX（接收）和TX（传输）队列用作Catalyst 8000V应用与AWS基础设施或硬件之间的连接点，管理网络流量的发送和接收方式。AWS根据实例类型控制每个接口可用的RX/TX队列的数量和速度。

要创建多个TX队列，Catalyst 8000V需要具有多个接口。当启用多个TX队列时，设备使用基于流5元组（源IP、目标IP、源端口、目标端口和协议）的散列方法来保持数据包流的顺序。 此散列决定每个流使用哪个TX队列。

用户可以使用连接到AWS实例的同一物理网络接口卡(NIC)在Catalyst 8000V上创建多个接口。这是通过配置环回接口或添加辅助IP地址来实现的。

使用多TXQ时，有多个传输队列来处理传出流量。在本例中，有12个TX队列（编号为0到11）。 此设置允许您单独监控每个队列，以查看是否有队列变满。

查看输出时，您可以看到TX Queue 8具有非常高的“满”计数器(56,406,998)，这意味着其缓冲区频繁填满。其他TX队列的“full”计数显示零，表示它们未拥塞。

```
Router#show platform hardware qfp active datapath infrastructure sw-cio
pmd b17a2f00 device Gi2
RX: pkts 9525 bytes 1229599 return 0 badlen 0
Out-of-credits: Hi 0 Lo 0
pkts/burst 1 cycl/pkt 560 ext_cycl/pkt 360
Total ring read 117322273, empty 117312792
TX: pkts 175116324 bytes 246208197526
pri-0: pkts 157 bytes 10238
pkts/send 1
pri-1: pkts 75 bytes 4117
pkts/send 1
pri-2: pkts 91 bytes 6955
pkts/send 1
pri-3: pkts 95 bytes 8021
pkts/send 1
pri-4: pkts 54 bytes 2902
pkts/send 1
pri-5: pkts 75 bytes 4082
pkts/send 1
pri-6: pkts 104 bytes 8571
pkts/send 1
pri-7: pkts 74 bytes 4341
pkts/send 1
pri-8: pkts 175115328 bytes 246208130411
pkts/send 2
pri-9: pkts 85 bytes 7649
pkts/send 1
pri-10: pkts 106 bytes 5784
pkts/send 1
pri-11: pkts 82 bytes 7267
pkts/send 1
Total: pkts/send 2 cycl/pkt 203
```

```
send 68548581 sendnow 175024880
forced 1039215617 poll 1155226129 thd_poll 0
blocked 2300918060 retries 68534370 mbuf alloc err 0
TX Queue 0: full 0 current index 0 hiwater 0
TX Queue 1: full 0 current index 0 hiwater 0
TX Queue 2: full 0 current index 0 hiwater 0
TX Queue 3: full 0 current index 0 hiwater 0
TX Queue 4: full 0 current index 0 hiwater 0
TX Queue 5: full 0 current index 0 hiwater 0
TX Queue 6: full 0 current index 0 hiwater 0
TX Queue 7: full 0 current index 0 hiwater 0
TX Queue 8: full 56406998 current index 224 hiwater 224 <<<<<<<<<<
TX Queue 9: full 0 current index 0 hiwater 0
TX Queue 10: full 0 current index 0 hiwater 0
TX Queue 11: full 0 current index 0 hiwater 0
```

监控TX队列的“满”计数器有助于识别任何传输队列是否过载。特定TX队列的“满”计数持续增加，指向设备承受压力的流量。解决此问题可能需要流量平衡、调整配置或扩展资源以提高性能。

超过指标

AWS在实例级别设置了某些网络限制，以确保不同实例大小的网络性能一致且优质。这些限制有助于为所有用户保持稳定的网络。

您可以在设备上使用show controllers命令检查这些限制和相关统计信息。输出包括许多计数器，但此处我们只关注用于监控网络性能的最重要计数器：

```
c8kv-2#sh control | inc exceed
<snipped>
bw_in_allowance_exceeded 0
bw_out_allowance_exceeded 0
pps_allowance_exceeded 0
conntrack_allowance_exceeded 0
linklocal_allowance_exceeded 0
<snipped>
```

现在，您可以深入了解这些计数器具体指什么：

- bw_in_allowance_exceeded:由于传入带宽超过实例限制而排队或丢弃的数据包数。
- bw_out_allowance_exceeded:由于传出带宽超出实例限制而排队或丢弃的数据包数。
- pps_allowance_exceeded:由于每秒数据包总数(PPS)超过实例限制而排队或丢弃的数据包数。
- conntrack_allowance_exceeded:跟踪的连接数达到实例类型允许的最大值。
- linklocal_allowance_exceeded:由于流向本地代理服务（如Amazon DNS、实例元数据服务和时间同步服务）的流量超过网络接口的PPS限制而丢弃的数据包数量。这不会影响自定义

DNS解析器。

这对您的C8000v性能意味着什么：

- 如果您注意到这些计数器不断增加并且遇到性能问题，这并不一定意味着C8000v路由器是问题所在。相反，它通常表示您使用的AWS实例已达到其容量限制。您可以检查AWS实例的规格，以确保其能够处理您的流量需求。

Microsoft Azure

在本节中，探索Microsoft Azure和Cisco C8000v虚拟路由器如何结合以在云中提供可扩展、安全且高性能的虚拟网络解决方案。

了解加速网络(AN)和数据包分段如何影响性能。以及检查对Microsoft Azure使用受支持的实例类型的重要性。

加速网络

在Microsoft Azure云中托管C8000v的性能问题中。一个无法忽视的方面是加速网络是否已启用。因为它极大地提高了路由器的性能。简而言之，加速网络在虚拟机（例如Cisco Catalyst 8000V VM）上启用单根I/O虚拟化(SR-IOV)。加速的网络路径会绕过虚拟交换机，提高网络流量的速度，改善网络性能，并降低网络延迟和抖动。

检查是否已启用加速网络的方法非常简单。即检查show controllers输出并验证是否存在特定计数器：

```
------------------ show controllers ------------------

GigabitEthernet1 - Gi1 is mapped to UIO on VXE
  rx_good_packets 6497723453
  tx_good_packets 14690462024
  rx_good_bytes 2271904425498
  tx_good_bytes 6276731371987




  rx_q0_good_packets 58576251
  rx_q0_good_bytes 44254667162
```

```
vf_rx_good_packets 6439147188
vf_tx_good_packets 14690462024
vf_rx_good_bytes 2227649747816
vf_tx_good_bytes 6276731371987
```

您正在查找的计数器是以vf开头的计数器，例如vf_rx_good_packets。如果您验证存在这些计数器，您可以绝对确保已启用加速网络。

Azure和分段

碎片化可能会对性能造成负面影响。影响性能的主要原因之一是数据包分段和重组的CPU/内存影响。当网络设备需要分段数据包时，它必须分配CPU/内存资源来执行分段。

重组数据包时也会发生同样的情况。网络设备必须存储所有分段，直到收到这些分段，这样它才能将其重组为原始数据包。

Azure不会使用加速网络处理分段的数据包。当VM收到分段数据包时，非加速路径会处理该数据包。因此，分段的数据包会错失加速网络的优势，例如延迟更短、抖动更小以及每秒数据包数更高。因此，建议尽可能避免支离破碎。

默认情况下，Azure会丢弃无序到达VM的分段数据包，这意味着这些数据包与来自源端点的传输序列不匹配。当数据包通过Internet或其他大型WAN传输时，可能会出现此问题。

Microsoft Azure支持的实例类型

根据思科标准，C8000v使用支持的实例类型非常重要。可以在Cisco Catalyst 8000V Edge软件安装和配置指南中找到。

这是因为该列表中的实例类型是C8KV正确测试的实例类型。现在，C8000v是否在不列出的实例类型上工作，有一个有效的问题？答案最有可能是肯定的。但是，当您对像性能问题这样的复杂问题进行故障排除时，不希望将其他未知因素添加到问题中。仅出于这一原因，思科TAC始终建议您保持受支持的实例类型。

其它资源

性能问题只有在当前发生时才能真正解决。然而，这很难被理解，因为这种情况可能随时发生。因

此，我们提供此EEM脚本。它有助于在数据包开始被丢弃和出现性能问题时捕获重要输出：

```
ip access-list extended TAC
permit ip host  host



permit ip host


        host



conf t
event manager applet CONNECTIONLOST1 authorization bypass
event track 100 state down maxrun 500
action 0010 syslog msg "Logging information to file bootflash:SLA-DROPS.txt and bootflash:FIASLA_Decode
action 0020 cli command "enable"
action 0021 cli command "term length 0"
action 0022 cli command "term exec prompt timestamp"
action 0023 cli command "term exec prompt expand"
action 0095 cli command "show clock | append bootflash:SLA-DROPS.txt"
action 0096 cli command "show platform hardware qfp active statistics drop detail | append bootflash:SL
action 0097 cli command "show logging | append bootflash:SLA-DROPS.txt"
action 0099 cli command "show interfaces summary | append bootflash:SLA-DROPS.txt"
action 0100 cli command "show interfaces | append bootflash:SLA-DROPS.txt"
action 0101 cli command "show platform hardware qfp active statistics drop clear"
action 0102 cli command "debug platform packet-trace packet 2048 fia-trace"
action 0103 cli command "debug platform packet-trace copy packet both"
action 0104 cli command "debug platform condition ipv4 access-list TAC both"
action 0105 cli command "debug platform condition start"
action 0106 cli command "show platform hardware qfp active data infrastructure sw-cio | append bootflas
action 0110 wait 60
action 0111 cli command "debug platform condition stop"
action 0112 cli command "show platform packet-trace packet all decode | append bootflash:FIASLA_Decode.
action 0120 cli command "show platform packet-trace statistics | append bootflash:FIASLA_Decode.txt"
action 0121 cli command "show platform packet-trace summary | append bootflash:FIASLA_Decode.txt"
action 0122 cli command "show platform hardware qfp active datapath utilization summary | append bootfl
action 0123 cli command "show platform hardware qfp active statistics drop detail | append bootflash:SL
action 0124 cli command "show platform hardware qfp active infrastructure bqs queue output default all
action 0125 cli command "show platform software status control-processor brief | append bootflash:SLA-D
action 0126 cli command "show platform hardware qfp active datapath infrastructure sw-pktmem | append b
action 0127 cli command "show platform hardware qfp active infrastructure punt statistics type per-caus
action 0128 cli command "show platform hardware qfp active statistics drop | append bootflash:SLA-DROPS
action 0129 cli command "show platform hardware qfp active infrastructure bqs queue output default all
action 0130 cli command "show platform hardware qfp active data infrastructure sw-hqf config 0 0 | appe
action 0131 cli command "show platform hardware qfp active feature lic-bw oversubscription | append boo
action 0132 cli command "show platform hardware qfp active data infrastructure sw-hqf config 0 0 | appe
action 0133 cli command "show platform hardware qfp active data infrastructure sw-cio | append bootflas
action 0134 cli command "show platform hardware qfp active data infrastructure sw-hqf sched | append bo
action 0135 cli command "show platform hardware qfp active data infrastructure sw-dist | append bootfla
action 0136 cli command "show platform hardware qfp active data infrastructure sw-nic | append bootflas
action 0137 cli command "show platform hardware qfp active data infrastructure sw-pktmem | append bootf
action 0138 cli command "show controllers | append bootflash:SLA-DROPS.txt"
action 0139 cli command "show platform hardware qfp active datapath pmd controllers | append bootflash:
action 0140 cli command "show platform hardware qfp active datapath pmd system | append bootflash:SLA-D
action 0141 cli command "show platform hardware qfp active datapath pmd static-if-config | append bootf
```

```
action 0150 cli command "clear platform condition all"
action 0151 cli command "clear platform packet-trace statistics"
action 0152 cli command "clear platform packet-trace configuration"
action 0153 cli command "show log | append bootflash:througput_levelinfoSLA.txt"
action 0154 cli command "show version | append bootflash:througput_levelinfoSLA.txt"
action 0155 cli command "show platform software system all | append bootflash:througput_levelinfoSLA.tx
action 0156 syslog msg "EEM script and FIA trace completed."
action 0180 cli command "conf t"
action 0181 cli command "no event manager applet CONNECTIONLOST1"
end
```

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。