使用單線命令對交換矩陣進行故障排除和驗證

目錄

簡介

本文描述了我們在交換矩陣上檢測整體問題的不同方法。

必要條件

需求

- 思科建議瞭解ACI

- 巴什語基礎知識

採用元件

本文件所述內容不限於特定軟體和硬體版本。

使用的裝置：

- 運行版本4.2(3)的Cisco ACI

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

背景資訊

用於獲取資訊的工具

- 用於替代：sed "s/<oldword >/<new wordk>/g" g定義它執行多次。
- 用於自始至終抓取線條：sed "/<beginning/,/end/p>"。將 — n(noprint)選項與/p列印標誌相結合，以複製grep的功能。
- Sed可以通過使用「；」使用多次，例如：將：sed s/<oldword >/<new word>/g;s/<oldword2 >/<new wordk2>/g"
- awk -F '<field_separator>' '{print $2}'在本特定示例中，您按定義的FIELD_SEPARATOR拆分行，並列印第2個分隔的區塊。兩個語法都做了完全相同的事情。
- awk '{ print $1, $2 }'列印每個輸入記錄的前兩個欄位，這兩個欄位之間有一個空格。
- sort | uniq 報告重複的行。按發生次數使用-c字首行。
- sort -nrk <column>對行進行最大排序。-n表示數字排序， -k表示鍵，以便我們可以修改列，如果要定義最低值，可以刪除 — r
- python -m json.tool以漂亮的格式顯示JSON。

所有單行命令的清單

僅獲取交換矩陣內枝葉的節點ID:

1. 在清單中：

```
APIC#acidiag fnvread | grep leaf | awk '{print $1}'
```

2. 在用逗號作為分隔符的行中：

```
APIC#acidiag fnvread | grep leaf | awk '{print $1}' | sed -z 's/\n/,/g;s/,$/\n/'
```

驗證是否有介面重設：

資訊在重置數最高的介面上排序。

```
APIC#moquery -c ethpm.PhysIf | egrep "dn|lastLinkStChg|resetCtr" | tr -d "\n" | sed "s/dn/\ndn/g;s/last
```

較慢選項

```
APIC#for leaf in `acidiag fnvread | grep leaf | awk '{print $1}'`; do echo; echo " -> leaf ID: $leaf  "
```

檢查最高分級介面：

若要尋找接收最多流量的位置：

查詢交換矩陣，查詢輸出吞吐量超過特定值(b)的所有介面。 m的值定義b是gb、mb還是kb。

要按gb篩選，請將m設定為125000000。要按mb篩選，請將m設定為125000。要按kb篩選，請將m設定為125。

```
APIC#bash
APIC#b=1; m=125000;b=$((b*m)); printf "%-65s %25s\n", "Node/Interface" "Bits/Second"; icurl 'http://loc
```

```
APIC#for leaf in `acidiag fnvread | grep leaf | awk '{print $1}'`; do echo; echo " -> leaf ID: $leaf  "
```

查詢所有STP拓撲更改：

1. 該命令將進入每個枝葉，並驗證是否有任何最近的拓撲更改以及哪些介面：

```
APIC#for node in `acidiag fnvread | grep leaf | awk '{print $1}'`; do echo; echo "node ID: $node "; fab
```

2. 該命令進入每個枝葉，並驗證PVRSTP更改的最大計數以及可見的介面：

```
APIC#for node in `acidiag fnvread | grep leaf | awk '{print $1}'`; do echo; echo "node ID: $node "; fab
```

確保是否存在多播RPF丟棄：

這需要在當時用於單個枝葉，並且它驗證所有RPF丟棄的所有已啟用的pim VRF:

```
APIC#for vrf in `show ip mroute summary vrf all | grep 'IP Multicast Routing Table for VR' | awk '{prin
```

驗證是否有太多資料包獲得Glean:

1. 交換矩陣ARP閃爍：

```
APIC#for leaf in `acidiag fnvread | grep leaf | awk '{print $1}'`; do echo; echo " -> leaf ID: $leaf  "
```

2. ARP收集收到的資料包：

```
APIC#for leaf in `acidiag fnvread | grep leaf | awk '{print $1}'`; do echo; echo " -> leaf ID: $leaf  "
```

QoS丟棄統計資訊：

驗證整個交換矩陣接收的QoS丟包：

```
 APIC#moquery -c qosmIfClass -f 'qosm.IfClass.RxDropPacketsCount!="0"' | egrep "RxDropPacketsCount|dn"
```

驗證整個交換矩陣的QoS傳輸丟棄：

```
APIC#moquery -c qosmIfClass -f 'qosm.IfClass.TxDropPacketsCount!="0"' | egrep "TxDropPacketsCount|dn" |
```

介面捨棄：

FCS錯誤（非堆積CRC錯誤）

```
APIC#moquery -c rmonDot3Stats -f 'rmon.Dot3Stats.fCSErrors>="1"' | egrep "dn|fCSErrors"
```

FCS +儲存的CRC錯誤

```
APIC#moquery -c rmonEtherStats -f 'rmon.EtherStats.cRCAlignErrors>="1"' | egrep "dn|cRCAlignErrors"
```

輸出緩衝區丟棄

```
APIC#moquery -c rmonEgrCounters -f 'rmon.EgrCounters.bufferdroppkts>="1"' | egrep "dn|bufferdroppkts"
```

輸出錯誤

```
APIC#moquery -c rmonIfOut -f 'rmon.IfOut.errors>="1"' | egrep "dn|errors"
```

清除所有介面計數器

用於獲取交換矩陣節點清單的命令

```
APIC# acidiag fnvread | egrep " active" | egrep "leaf|spine" | awk '{print $1}' | sed -e 'H;${x;s/\n/,/
101,102,103,204,205,206,301,1001,1002,2001,2002
```

用於清除以前清單的計數器的命令

```
APIC# fabric 101,102,103,204,205,206,301,1001,1002,2001,2002 clear counters interface all
```

BGP會話問題：

檢查交換矩陣底層上是否有任何BGP作業階段出現問題

```
APIC#moquery -c bgpPeerEntry -f 'bgp.PeerEntry.operSt!="established" and  bgp.PeerEntry.dn*"overlay-1"'
```

檢查任何BGP作業階段

```
APIC#moquery -c bgpPeerEntry -f 'bgp.PeerEntry.operSt!="established" | egrep "dn|operSt" | tr -d "\n" |
```

OSPF會話問題：

識別未處於完全狀態的會話。

```
APIC#moquery -c ospfAdjEp -f 'ospf.AdjEp.operSt!="full"' | egrep "dn|peerIp" | tr -d '\n' | sed "s/dn
```

識別不斷擺動的作業階段，並將資訊排序到最高位數：

```
APIC#moquery -c ospfAdjStats -f 'ospf.AdjStats.stChgCnt!="0"' | egrep "dn|stChgCnt" | tr -d "\n" | tr -
```

被傳送到CPU的主資料包

⚠ 考慮：此命令調試500個資料包。對於較小的金額，請修改 — c後的數字。

```
lEAF#tcpdump -i kpm_inb -c 500 > /tmp/cpu-dp.txt
```

```
LEAF#cat /tmp/cpu-dp.txt | grep IP | awk '{print $3 , $4 , $5}' | grep -v $HOSTNAME | awk -F ':' '{print
```

直接連線，而不建立新檔案：

```
LEAF#tcpdump -i kpm_inb -c 500 | grep IP | awk '{print $3 , $4 , $5}' | grep -v $HOSTNAME | awk -F ':'
```

從部署所有特定合約關係的apic檢查整個交換矩陣

```
APIC#moquery -c actrlRule -f 'actrl.Rule.sPcTag=="32783" and actrl.Rule.dPcTag=="46" and actrl.Rule.sco
```

檢查封裝已在何處部署並獲得相應的epg

```
APIC#moquery -c l2CktEp -f 'l2.CktEp.encap=="vlan-3018"'
```

交換矩陣中所有節點的記憶體利用率：

```
APIC#bash
APIC# clear ; echo -e "Node ID\t\tFree Memory\tUsed Memory" ; moquery -c procSysMemHist15min -f 'proc.S
```

驗證istack上的丟棄（異常資料包被點選）

檢查TX丟棄。使用sort -nk 6 -r的命令：

```
 APIC# cat istack_debug | egrep "Protocol:|x_pkts_dropped" | tr -d "\n" | sed "s/Protocol/\nProtocol/g"
```

合約驗證

複查特定合約的所有關係。使用指令碼替換合約和租戶的名稱：

```
APIC# CONTRACT='brc-<contract-name>'
APIC# TN='<tenant>'
#CHECK CONSUMERS
#To get the count of epgs consuming a contract (excluding vzany consumer):
APIC#icurl -g 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=subtree&tar

#To list all epg objects consuming a contract (excluding vzany consumers):
APIC#icurl -g 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=subtree&tar

#To get the count of vzanys consuming a contract:
APIC#icurl 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=children&targe

#To list all vzany objects consuming a contract:
APIC#icurl 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=children&targe

#CHECK PROVIDERS
#To get the count of epgs providing a contract (excluding vzany consumer):
APIC#icurl -g 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=subtree&tar

#To list all epg objects providing a contract (excluding vzany consumers):
```

```
APIC#icurl -g 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=subtree&tar

#To get the count of vzanys providing a contract:
APIC#icurl 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=children&targe

#To list all vzany objects providing a contract:
APIC#icurl 'http://localhost:7777/api/node/mo/uni/tn-'$TN'/'$CONTRACT'.json?query-target=children&targe
```

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。