

排除Nexus 7000高CPU使用率故障

目錄

[簡介](#)

[Nexus 7000平台上的CPU使用率](#)

[用於監控進程和CPU的命令和指令碼](#)

指令

[show processes命令](#)

[show system resources命令](#)

[show processes cpu命令](#)

[show processes cpu history命令](#)

[show process cpu detail 指令](#)

簡介

本文檔介紹用於監視Cisco Nexus 7000系列平台上的CPU使用情況和排除CPU使用率高問題的過程。

Nexus 7000平台上的CPU使用率

Nexus 7000平台是一個基於Linux的系統，它有一個搶佔式排程程式，允許對所有進程的CPU資源進行公平訪問。

與Cisco Catalyst 6500系列不同，它沒有單獨的路由處理器(RP)和交換機處理器(SP)。

- 管理引擎1具有雙核處理器。
- 管理引擎2採用四核處理器。
- 管理引擎2E有兩個四核處理器。

Cisco NX-OS作業系統利用搶先式CPU多工處理，因此進程可以利用空閒CPU更快地完成任務。

因此，history選項報告不一定表示問題的可能的CPU峰值。但是，如果平均CPU使用率與特定網路的基準CPU使用率正常值相比保持較高水準，請調查高CPU使用率。

已啟用預設硬體速率限制器(HWRL)和預設控制平面策略(CoPP)，以幫助保護Nexus 7000平台上的管理引擎帶內介面。

命令和示例EEM指令碼基於Nexus 7000 6.1版及更低版本，可能會在未來版本中更改。

用於監控進程和CPU的命令和指令碼

指令

[Cisco CLI Analyzer \(僅供已註冊客戶使用 \) 支援某些 show 指令](#)。使用 Cisco CLI Analyzer 檢視

show 指令輸出的分析。

show processes命令

使用此命令可顯示有關活動進程的資訊。

```
switch# show processes
```

| PID | State | PC | Start_cnt | TTY | Type | Process |
|-----|-------|----------|-----------|-----|------|-------------|
| 1 | S | 41520eb8 | 1 | - | 0 | init |
| 2 | S | 0 | 1 | - | 0 | kthreadd |
| 3 | S | 0 | 1 | - | 0 | migration/0 |
| 4 | S | 0 | 1 | - | 0 | ksoftirqd/0 |
| 5 | S | 0 | 1 | - | 0 | watchdog/0 |
| 6 | S | 0 | 1 | - | 0 | migration/1 |
| 7 | S | 0 | 1 | - | 0 | ksoftirqd/1 |
| 8 | S | 0 | 1 | - | 0 | watchdog/1 |
| 9 | S | 0 | 1 | - | 0 | events/0 |
| 10 | S | 0 | 1 | - | 0 | events/1 |
| 11 | S | 0 | 1 | - | 0 | khelper |
| 12 | S | 0 | 1 | - | 0 | kblockd/0 |

| | |
|--------|------------------------------------|
| 欄位 | 說明 |
| PID | 進程ID |
| 狀態 | 進程狀態 |
| PC | 十六進位制格式的當前程式計數器 |
| 啟動_cnt | 進程已啟動或重新啟動的次數 |
| TTY | 控制進程的終端。連字元(-)通常表示守護程式不在任何特定終端上運行。 |
| 流程 | 進程的名稱 |

| | |
|------|------------------|
| 進程狀態 | 說明 |
| D | 不間斷睡眠 (通常為I/O) |
| R | 可運行 (在運行隊列上) |
| S | 睡覺 |
| T | 已跟蹤或已停止 |
| Z | 已停用 (殭屍) 進程 |
| NR | 未運行 |
| ER | 預期正在運行，但當前未運行 |

show system resources命令

使用此命令可顯示系統相關的CPU和記憶體統計資訊。

```
switch#show system resources
Load average: 1 minute: 0.36 5 minutes: 0.39 15 minutes: 0.44
Processes : 1068 total, 1 running
CPU states : 0.5% user, 5.5% kernel, 94.0% idle
Memory usage: 8245436K total, 3289920K used, 4955516K free
Current memory status: OK
```

| 欄位 | 說明 |
|---------|--|
| 載入 | 正在運行的進程數。平均值反映了過去1、5和15分鐘的系統負載。 |
| 流程 | 系統中的進程數以及發出命令時實際運行的進程數。 |
| CPU狀態 | 使用者模式、核心模式和上一秒空閒時間中的CPU使用率。對於雙核 Supervisor，CPU平均分佈在兩個核心上。 |
| 記憶體使用情況 | 總記憶體、已用記憶體、可用記憶體、用於緩衝區的記憶體和用於快取記憶體的記憶體（以千位元組為單位）。緩衝區和快取記憶體包含在已用記憶體統計資訊中。 |

show processes cpu命令

使用以下命令以顯示處理層級的CPU使用率：

```
switch#show processes cpu | ex 0.0
```

```
PID Runtime(ms) Invoked uSecs 1Sec Process
-----
```

```
26 66399 269718 246 0.9% kide/1
2908 115550 11310 10216 2.9% platform
3223 7248 9208 787 0.9% R2D2_usd
```

```
CPU util : 1.0% user, 3.0% kernel, 96.0% idle
```

Please note that only processes from the requested vdc are shown above

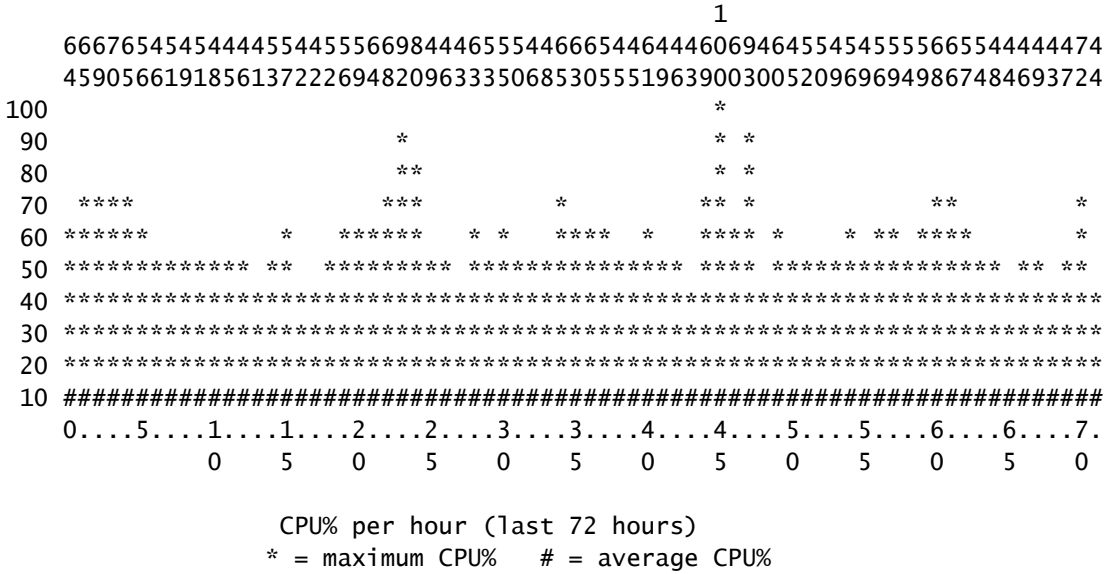
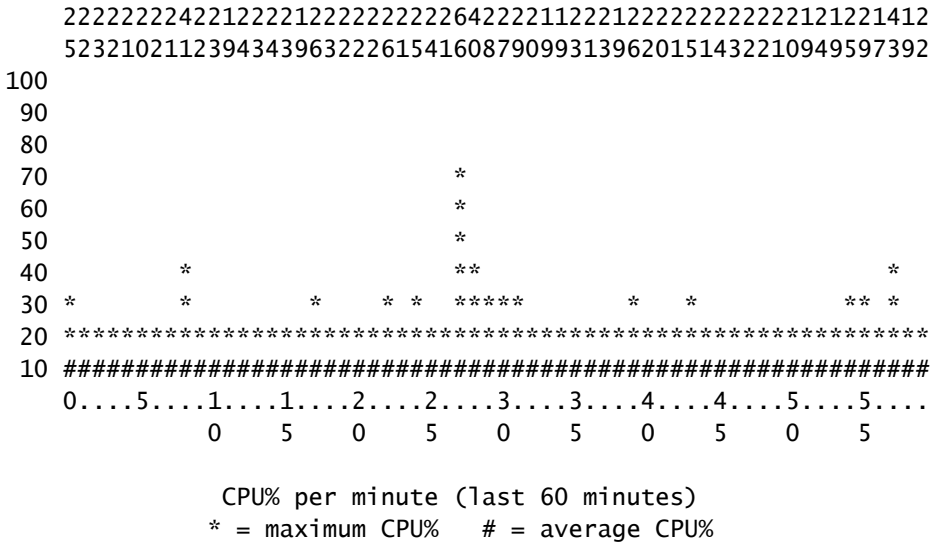
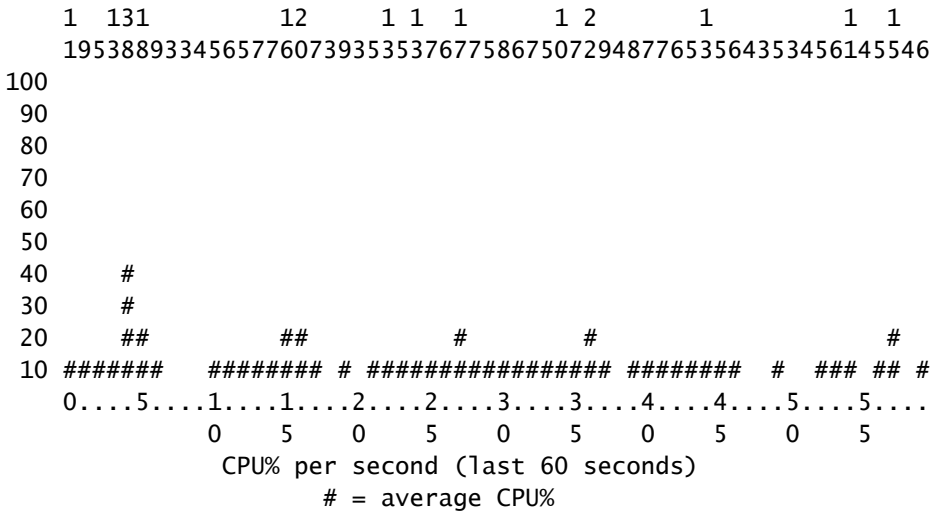
| 欄位 | 說明 |
|------------|-----------------------|
| 運行時 (毫秒) | 進程已使用的CPU時間 (毫秒) |
| 已呼叫 | 已呼叫進程的次數 |
| 秒 | 每個進程呼叫的平均CPU時間 (微秒) |
| 1秒 | 最近一秒的CPU使用率 |

要查詢屬於特定進程ID(PID)的所有執行緒的CPU使用情況，請使用show process cpu detail <pid>命令，該命令在NX-OS版本6.2x中提供。

show processes cpu history命令

使用此命令可以顯示過去60秒、60分鐘和72小時的CPU使用率。請務必檢查平均CPU使用率(#)和峰值(*)。

switch# show processes cpu history



show process cpu detail <pid> 指令

此命令 (在6.2版中新增) 顯示屬於特定PID的所有執行緒的CPU使用資訊。

```
switch# show processes cpu sorted | grep cli
```

| | | | | | | | | |
|------|-------|-------|------|------|------|------|---|-------------|
| 3965 | 23734 | 17872 | 1328 | 0.0% | 0.1% | 0.7% | - | cli |
| 4024 | 3047 | 1256 | 2426 | 0.0% | 0.0% | 0.0% | - | diagclient |
| 4094 | 787 | 258 | 3052 | 0.0% | 0.0% | 0.0% | - | cardclient |
| 4728 | 227 | 209 | 1088 | 0.0% | 0.0% | 0.0% | - | port_client |
| 4729 | 1351 | 499 | 2708 | 0.0% | 0.0% | 0.0% | - | statsclient |
| 4730 | 2765 | 550 | 5028 | 0.0% | 0.0% | 0.0% | - | xbar_client |

```
switch# show processes cpu sorted | grep clis
```

| | | | | | | | | |
|------|-------|-------|------|------|------|------|---|-----|
| 3965 | 23734 | 17872 | 1328 | 0.0% | 0.1% | 0.7% | - | cli |
|------|-------|-------|------|------|------|------|---|-----|

```
switch# show process cpu detailed 3965
```

CPU utilization for five seconds: 3%/3%; one minute: 0%; five minutes: 1%

| PID | Runtime(ms) | Invoked | uSecs | 5Sec | 1Min | 5Min | TTY | Process |
|------|-------------|---------|-------|------|------|------|-----|-----------------|
| 3965 | 23734 | 17873 | 1327 | 0.0% | 0.1% | 0.6% | - | cli |
| 4227 | 45 | 334 | 135 | 0.0% | 0.0% | 0.0% | - | clis:clis-cli-t |
| 4228 | 24 | 153 | 162 | 0.0% | 0.0% | 0.0% | - | clis:clis-nvdb- |
| 4760 | 75 | 224 | 335 | 0.0% | 0.0% | 0.0% | - | clis:clis-seria |

```
switch# show processes cpu sorted | grep netstack
```


| | | | | | | | | |
|------|-----|-----|-----|------|------|------|---|----------|
| 4133 | 353 | 892 | 395 | 0.0% | 0.0% | 0.0% | - | netstack |
|------|-----|-----|-----|------|------|------|---|----------|

```
switch# show process cpu detailed 4133
```

CPU utilization for five seconds: 5%/5%; one minute: 1%; five minutes: 1%

| PID | Runtime(ms) | Invoked | uSecs | 5Sec | 1Min | 5Min | TTY | Process |
|------|-------------|---------|-------|------|------|------|-----|-----------------|
| 4133 | 353 | 892 | 395 | 0.0% | 0.0% | 0.0% | - | netstack |
| 4145 | 322 | 6492 | 49 | 0.0% | 0.0% | 0.0% | - | netstack:active |
| 4151 | 239 | 247 | 971 | 0.0% | 0.0% | 0.0% | - | netstack:ip-sys |
| 4153 | 0 | 3 | 162 | 0.0% | 0.0% | 0.0% | - | netstack:mp1sda |
| 4155 | 2 | 3 | 717 | 0.0% | 0.0% | 0.0% | - | netstack:mp1sct |
| 4163 | 0 | 2 | 240 | 0.0% | 0.0% | 0.0% | - | netstack:ipv6-d |
| 4164 | 97 | 957 | 101 | 0.0% | 0.0% | 0.0% | - | netstack:netsta |
| 4166 | 15 | 628 | 25 | 0.0% | 0.0% | 0.0% | - | netstack:ip-sys |
| 4167 | 0 | 3 | 224 | 0.0% | 0.0% | 0.0% | - | netstack:ip-pm- |
| 4170 | 1 | 12 | 154 | 0.0% | 0.0% | 0.0% | - | netstack:ip-uri |
| 4171 | 9 | 30 | 323 | 0.0% | 0.0% | 0.0% | - | netstack:ip-ipc |
| 4173 | 0 | 5 | 167 | 0.0% | 0.0% | 0.0% | - | netstack:ip-ipc |
| 4175 | 0 | 2 | 305 | 0.0% | 0.0% | 0.0% | - | netstack:ip-ret |
| 4176 | 12 | 7 | 1838 | 0.0% | 0.0% | 0.0% | - | netstack:ip-ppf |
| 4178 | 4 | 15 | 289 | 0.0% | 0.0% | 0.0% | - | netstack:ipv6-c |
| 4179 | 41 | 445 | 93 | 0.0% | 0.0% | 0.0% | - | netstack:disp |
| 4180 | 0 | 6 | 98 | 0.0% | 0.0% | 0.0% | - | netstack:worker |
| 4181 | 33 | 501 | 66 | 0.0% | 0.0% | 0.0% | - | netstack:worker |
| 4182 | 0 | 2 | 232 | 0.0% | 0.0% | 0.0% | - | netstack:worker |
| 4183 | 0 | 2 | 227 | 0.0% | 0.0% | 0.0% | - | netstack:worker |
| 4184 | 0 | 3 | 152 | 0.0% | 0.0% | 0.0% | - | netstack:worker |
| 4185 | 0 | 2 | 278 | 0.0% | 0.0% | 0.0% | - | netstack:worker |
| 4186 | 0 | 2 | 254 | 0.0% | 0.0% | 0.0% | - | netstack:worker |
| 4187 | 0 | 3 | 168 | 0.0% | 0.0% | 0.0% | - | netstack:worker |
| 4188 | 0 | 2 | 266 | 0.0% | 0.0% | 0.0% | - | netstack:worker |
| 4189 | 0 | 2 | 248 | 0.0% | 0.0% | 0.0% | - | netstack:worker |
| 4190 | 0 | 2 | 254 | 0.0% | 0.0% | 0.0% | - | netstack:worker |
| 4191 | 0 | 3 | 201 | 0.0% | 0.0% | 0.0% | - | netstack:worker |

| | | | | | | | | |
|------|----|-----|-----|------|------|------|---|-----------------|
| 4192 | 0 | 2 | 258 | 0.0% | 0.0% | 0.0% | - | netstack:worker |
| 4193 | 0 | 7 | 111 | 0.0% | 0.0% | 0.0% | - | netstack:worker |
| 4194 | 0 | 8 | 78 | 0.0% | 0.0% | 0.0% | - | netstack:worker |
| 4195 | 0 | 2 | 313 | 0.0% | 0.0% | 0.0% | - | netstack:worker |
| 4196 | 15 | 632 | 23 | 0.0% | 0.0% | 0.0% | - | netstack:ptacti |
| 4197 | 0 | 5 | 120 | 0.0% | 0.0% | 0.0% | - | netstack:tcp_ip |
| 4198 | 4 | 11 | 390 | 0.0% | 0.0% | 0.0% | - | netstack:ipv6-m |
| 4199 | 0 | 3 | 240 | 0.0% | 0.0% | 0.0% | - | netstack:ipv6-c |
| 4200 | 0 | 1 | 561 | 0.0% | 0.0% | 0.0% | - | netstack:ipv6-c |
| 4201 | 0 | 3 | 246 | 0.0% | 0.0% | 0.0% | - | netstack:icmpv6 |
| 4513 | 0 | 5 | 112 | 0.0% | 0.0% | 0.0% | - | netstack:ipv6-m |
| 4514 | 0 | 2 | 291 | 0.0% | 0.0% | 0.0% | - | netstack:ipv6-m |

 註：所有流程資訊均基於NX-OS中的「proc」。在NX-OS中，所有執行緒共用由任何其他執行緒分配的記憶體，因此不可能顯示每個執行緒的資訊。

show system internal processes cpu命令

此命令等效於Linux中的top命令，可即時檢視處理器活動。

```
switch# show system internal processes cpu
```

```
top - 23:51:41 up 51 min, 3 users, load average: 0.56, 0.49, 0.46
Tasks: 433 total, 1 running, 431 sleeping, 0 stopped, 1 zombie
Cpu(s): 5.9%us, 7.8%sy, 0.0%ni, 81.9%id, 3.6%wa, 0.1%hi, 0.6%si, 0.0%st
Mem: 8245436k total, 3531776k used, 4713660k free, 5360k buffers
Swap: 0k total, 0k used, 0k free, 1458188k cached
```

```
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
3589 svc-isan 25 5 112m 8864 4572 S 5.7 0.1 0:21.60 stats_client
10881 sjlan 20 0 3732 1648 1140 R 3.8 0.0 0:00.04 top
26 root 20 0 0 0 0 S 1.9 0.0 1:07.07 kide/1
3280 root -2 0 101m 6104 3680 S 1.9 0.1 0:32.57 octopus
3570 root 20 0 123m 19m 6456 S 1.9 0.2 0:06.07 diag_port_lb
5151 root 20 0 205m 45m 9.8m S 1.9 0.6 0:02.61 netstack
1 root 20 0 1988 604 524 S 0.0 0.0 0:03.75 init
2 root 15 -5 0 0 0 S 0.0 0.0 0:00.00 kthreadd
3 root RT -5 0 0 0 S 0.0 0.0 0:00.00 migration/0
4 root 15 -5 0 0 0 S 0.0 0.0 0:00.61 ksoftirqd/0
5 root -2 -5 0 0 0 S 0.0 0.0 0:00.06 watchdog/0
6 root RT -5 0 0 0 S 0.0 0.0 0:00.00 migration/1
7 root 15 -5 0 0 0 S 0.0 0.0 0:04.80 ksoftirqd/1
```

| | |
|------|-------------|
| 欄位 | 說明 |
| PID | 進程ID |
| 使用者 | 擁有進程的使用者的名稱 |
| 公關 | 分配給進程的優先順序 |
| NI | 此流程有很好的價值 |
| VIRT | 進程使用的虛擬記憶體量 |

| | |
|------|--|
| RES | 進程正在使用的物理RAM大小 (駐留大小) (千位元組) |
| SHR | 進程使用的共用記憶體量 |
| S | 進程的狀態。可能的值包括： <ul style="list-style-type: none"> • D — 不間斷睡眠 • R — 正在運行 • S — 睡眠 • T — 已跟蹤或已停止 • Z — 殭屍 |
| %CPU | 進程使用的CPU時間百分比 |
| %MEM | 進程使用的可用物理RAM的百分比 |
| 時間+ | 進程自啟動以來已消耗的CPU時間總量 |
| 指令 | 為啟動進程而輸入的命令的名稱 |

「{#seconds} | no-more」選項允許每天自動執行命令#seconds直到輸入Ctrl-C。以下是輸出範例：

```
<#root>
```

```
switch# show system internal processes cpu
```

```
5 | no-more
```

```
top - 17:31:12 up 4 days, 18:31, 3 users, load average: 0.52, 0.40, 0.32
Tasks: 449 total, 3 running, 446 sleeping, 0 stopped, 0 zombie
Cpu(s): 3.5%us, 4.5%sy, 0.0%ni, 91.2%id, 0.1%wa, 0.1%hi, 0.5%si, 0.0%st
Mem: 8245436k total, 4192740k used, 4052696k free, 27644k buffers
Swap: 0k total, 0k used, 0k free, 1919612k cached
  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 2908 root        20   0  112m 8516 5516 S   7.5   0.1 264:44.25 pfm
31487 sjlan       20   0   3732 1652 1140 R   5.6   0.0  0:00.05 top
 3059 svc-isan   20   0 80288 7536 4440 S   3.8   0.1  65:44.59 diagmgr
 3192 root        20   0   334m  47m  11m S   1.9   0.6 25:36.52 netstack
 3578 svc-isan   20   0   118m  13m 6952 S   1.9   0.2 24:57.36 stp
 5119 svc-isan   20   0   139m  14m 7028 S   1.9   0.2  3:48.60 urib
 5151 root        20   0   209m  46m  11m S   1.9   0.6 38:53.39 netstack
 5402 svc-isan   20   0   117m  15m 9140 S   1.9   0.2 36:07.13 stp
 6175 svc-isan   20   0   118m  16m 9580 S   1.9   0.2 47:09.41 stp
    1 root        20   0   1988   604  524 S   0.0   0.0  0:06.51 init
    2 root        15  -5     0     0     0 S   0.0   0.0  0:00.00 kthreadd
    3 root         RT  -5     0     0     0 S   0.0   0.0  0:00.08 migration/0
    4 root        15  -5     0     0     0 S   0.0   0.0  1:07.77 ksoftirqd/0
```

```
top - 17:31:18 up 4 days, 18:31, 3 users, load average: 0.48, 0.39, 0.32
Tasks: 449 total, 1 running, 448 sleeping, 0 stopped, 0 zombie
Cpu(s): 3.5%us, 4.5%sy, 0.0%ni, 91.2%id, 0.1%wa, 0.1%hi, 0.5%si, 0.0%st
Mem: 8245436k total, 4192592k used, 4052844k free, 27644k buffers
Swap: 0k total, 0k used, 0k free, 1919612k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 2908 root        20   0  112m 8516 5516 S   7.5   0.1 264:44.47 pfm
31490 sjlan       20   0   3732 1656 1140 R   3.8   0.0  0:00.04 top
    1 root        20   0   1988   604  524 S   0.0   0.0  0:06.51 init
```

```

 2 root      15  -5   0   0   0 S  0.0  0.0   0:00.00 kthreadd
 3 root      RT  -5   0   0   0 S  0.0  0.0   0:00.08 migration/0
 4 root      15  -5   0   0   0 S  0.0  0.0   1:07.77 ksoftirqd/0
 5 root      -2  -5   0   0   0 S  0.0  0.0   0:13.74 watchdog/0
 6 root      RT  -5   0   0   0 S  0.0  0.0   0:00.10 migration/1
 7 root      15  -5   0   0   0 S  0.0  0.0   0:54.47 ksoftirqd/1
 8 root      -2  -5   0   0   0 S  0.0  0.0   0:00.20 watchdog/1
 9 root      15  -5   0   0   0 S  0.0  0.0   0:02.94 events/0
10 root      15  -5   0   0   0 S  0.0  0.0   0:02.58 events/1
11 root      15  -5   0   0   0 S  0.0  0.0   0:00.00 khelper
top - 17:31:23 up 4 days, 18:31,  3 users,  load average: 0.44, 0.39, 0.32
Tasks: 449 total,  1 running, 448 sleeping,  0 stopped,  0 zombie
Cpu(s):  3.5%us,  4.5%sy,  0.0%ni, 91.2%id,  0.1%wa,  0.1%hi,  0.5%si,  0.0%st
Mem:   8245436k total, 4192584k used, 4052852k free,  27644k buffers
Swap:      0k total,    0k used,    0k free, 1919612k cached

```

```

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
31493 sjlan    20   0  3732 1656 1140  R   3.8   0.0    0:00.04  top
 5004 svc-isan 20   0  118m 13m  6852  S   1.9   0.2   41:35.81  stp
10337 svc-isan 20   0  133m 11m  7948  S   1.9   0.1    1:42.81  mcecm
   1 root     20   0  1988  604   524  S   0.0   0.0    0:06.51  init
   2 root     15  -5    0    0    0 S   0.0   0.0    0:00.00  kthreadd
   3 root     RT  -5    0    0    0 S   0.0   0.0    0:00.08  migration/0
   4 root     15  -5    0    0    0 S   0.0   0.0    1:07.77  ksoftirqd/0
   5 root     -2  -5    0    0    0 S   0.0   0.0    0:13.74  watchdog/0
   6 root     RT  -5    0    0    0 S   0.0   0.0    0:00.10  migration/1
   7 root     15  -5    0    0    0 S   0.0   0.0    0:54.47  ksoftirqd/1
   8 root     -2  -5    0    0    0 S   0.0   0.0    0:00.20  watchdog/1
   9 root     15  -5    0    0    0 S   0.0   0.0    0:02.94  events/0

```

```

10 root     15  -5    0    0    0 S   0.0   0.0    0:02.58  events/1
top - 17:31:29 up 4 days, 18:31,  3 users,  load average: 0.41, 0.38, 0.32
Tasks: 449 total,  1 running, 448 sleeping,  0 stopped,  0 zombie
Cpu(s):  3.5%us,  4.5%sy,  0.0%ni, 91.2%id,  0.1%wa,  0.1%hi,  0.5%si,  0.0%st
Mem:   8245436k total, 4192708k used, 4052728k free,  27644k buffers
Swap:      0k total,    0k used,    0k free, 1919616k cached

```

show system internal sysmgr service pid <pid>命令

使用此命令可以按PID顯示進程/服務的其他詳細資訊，如重新啟動時間、崩潰狀態和當前狀態。

```

switch# show system internal processes cpu
top - 17:37:26 up 4 days, 18:37,  3 users,  load average: 0.16, 0.35, 0.33
Tasks: 450 total,  2 running, 448 sleeping,  0 stopped,  0 zombie
Cpu(s):  3.5%us,  4.5%sy,  0.0%ni, 91.2%id,  0.1%wa,  0.1%hi,  0.5%si,  0.0%st
Mem:   8245436k total, 4193248k used, 4052188k free,  27668k buffers
Swap:      0k total,    0k used,    0k free, 1919664k cached
  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 2908 root     20   0  112m 8516 5516  S   7.5   0.1  264:58.67  pfm
31710 sjlan    20   0  3732 1656 1140  R   3.8   0.0    0:00.04  top
 3192 root     20   0  334m 47m  11m  S   1.9   0.6   25:38.39  netstack
 3578 svc-isan 20   0  118m 13m  6952  S   1.9   0.2   24:59.08  stp
 5151 root     20   0  209m 46m  11m  S   1.9   0.6   38:55.52  netstack
 5402 svc-isan 20   0  117m 15m  9140  S   1.9   0.2   36:09.08  stp
 5751 root     20   0  209m 46m  10m  S   1.9   0.6   41:20.58  netstack
 6098 svc-isan 20   0  151m 15m  6188  S   1.9   0.2    3:58.40  mrrib
 6175 svc-isan 20   0  118m 16m  9580  S   1.9   0.2   47:12.00  stp

```




```
1 root      20   0 1988 604 524 S 0.0 0.0 0:06.52 init
2 root      15  -5   0   0   0 S 0.0 0.0 0:00.00 kthreadd
3 root      RT  -5   0   0   0 S 0.0 0.0 0:00.08 migration/0
4 root      15  -5   0   0   0 S 0.0 0.0 1:07.83 ksoftirqd/0
```

```
switch# show system internal sysmgr service pid 2908
Service "Platform Manager" ("platform", 5):
  UUID = 0x18, PID = 2908, SAP = 39
  State: SRV_STATE_HANDSHAKED (entered at time Mon Oct 15 23:03:45 2012).
  Restart count: 1
  Time of last restart: Mon Oct 15 23:03:44 2012.
  The service never crashed since the last reboot.
  Tag = N/A
  Plugin ID: 0
```

EEM指令碼示例

以下是捕獲間歇性高CPU使用率的示例指令碼。使用的值以及發出的命令可根據要求進行修改：

```
event manager applet HIGH-CPU
event snmp oid 1.3.6.1.4.1.9.9.109.1.1.1.1.6.1 get-type exact entry-op ge
  entry-val 80 exit-val 30 poll-interval 5
action 1.0 syslog msg High CPU hit $_event_pub_time
action 2.0 cli enable
action 3.0 cli show clock >> bootflash:high-cpu.txt
action 4.0 cli show processes cpu sort >> bootflash:high-cpu.txt
```

 註：必須定義「exit-val」。當指令碼收集資料時，會增加CPU利用率。exit-val的值可確保指令碼不會在無限循環中運行。

高CPU使用率由 進程或流量

監控CPU使用率時，沒有進程與中斷CPU使用率(如Cisco IOS[®]軟體平台上的情況)。要快速確定CPU使用率高的原因，請使用[show system internal processes cpu](#)命令。流量觸發的高CPU使用率很可能導致Netstack以及其他功能和進程(如地址解析協定(ARP)和網際網路組管理協定(IGMP))運行較高。

進程導致CPU使用率高

根據導致CPU使用率較高的進程和問題，可能需要捕獲特定命令。以下各節介紹有用的方法。

show system internal <feature> mem-stats/memstats | 大命令中

使用此命令可顯示進程的記憶體分配；使用「in Grand」選項可監視總記憶體。記憶體洩漏可能導致進程行為不當，從而導致CPU使用率高。

Ethalyzer

使用Ethalyzer監控到CPU的流量。

debug指令

 附註：使用 debug 指令之前，請先參閱有關 Debug 指令的重要資訊。在生產交換機上明智地使用debug命令以避免服務中斷。

儘可能使用debug logfile命令將輸出指向指定的檔案，並避免鎖定會話以填充系統日誌。以下是偵錯簡易網路管理通訊協定(SNMP)的範例：

```
switch# debug logfile snmpdebug
switch# debug snmp all
switch# show debug logfile snmpdebug
2012 Oct 17 23:53:25.905914 snmpd: SDWRAP message Successfully processed
2012 Oct 17 23:53:25.906162 snmpd: Src: 0x00000501/23852 Dst: 0x00000501/28 ID
      : 0x006E3C9B Size: 276 [REQ] Opc: 182 (MTS_OPC_DEBUG_WRAP_MSG) RR: 0x006E3C9B
      HA_SEQNO: 0x00000000 TS: 0x10ADFFA1666FC REJ:0 SYNC:0 OPTIONS:0x0
2012 Oct 17 23:53:25.906208 snmpd: 01 00 00 00 E7 03 00 00 00 00 00 00 00 00 00 00
2012 Oct 17 23:53:25.906225 snmpd: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2012 Oct 17 23:53:25.906239 snmpd: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2012 Oct 17 23:53:25.906255 snmpd: FF FF FF FF 2F 64 65 76 2F 70 74 73 2F 30 00 00
2012 Oct 17 23:53:25.906271 snmpd: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

switch# show log last 10
2012 Oct 17 17:51:06 SITE1-AGG1 %ETHPORT-5-IF_TX_FLOW_CONTROL: Interface
      Ethernet10/10, operational Transmit Flow Control state changed to off
2012 Oct 17 17:51:09 SITE1-AGG1 %ETH_PORT_CHANNEL-5-PORT_SUSPENDED:
      Ethernet10/10: Ethernet10/10 is suspended
2012 Oct 17 17:51:51 SITE1-AGG1 last message repeated 1 time
2012 Oct 17 17:51:51 SITE1-AGG1 %ETHPORT-5-IF_DOWN_LINK_FAILURE:
      Interface Ethernet10/10 is down (Link failure)
2012 Oct 17 17:51:52 SITE1-AGG1 %ETHPORT-5-SPEED: Interface Ethernet10/10,
      operational speed changed to 10 Gbps
2012 Oct 17 17:51:52 SITE1-AGG1 %ETHPORT-5-IF_DUPLEX: Interface
      Ethernet10/10, operational duplex mode changed to Full
2012 Oct 17 17:51:52 SITE1-AGG1 %ETHPORT-5-IF_RX_FLOW_CONTROL: Interface
      Ethernet10/10, operational Receive Flow Control state changed to off
2012 Oct 17 17:51:52 SITE1-AGG1 %ETHPORT-5-IF_TX_FLOW_CONTROL: Interface
      Ethernet10/10, operational Transmit Flow Control state changed to off
2012 Oct 17 17:51:55 SITE1-AGG1 %ETH_PORT_CHANNEL-5-PORT_UP: port-channel11:
      Ethernet10/10 is up
2012 Oct 17 17:51:56 SITE1-AGG1 %ETHPORT-5-IF_UP: Interface Ethernet10/10
      is up in mode trunk
```

儘可能使用debug-filter命令以最小化生產系統上的輸出。例如，封包遺失會導致單向連結偵測(UDLD)空回應：

```
switch# debug logfile test size 1000000
```

```
switch# debug-filter pktmgr direction inbound
switch# debug-filter pktmgr dest-mac 0100.0ccc.cccc
switch# debug pktmgr client uuid 376
switch# debug pktmgr frame
switch# debug pktmgr pkt-errors
```

```
switch# debug-filter ?
fabricpath  Debug fabricpath events
ip          IP events
ipv6       IPv6 events
l2pt       L2 Protocol Tunneling events
mpls       MPLS events
pktmgr     Pm debug-filter
routing    Routing events
```

流量導致CPU使用率高

當流量導致CPU使用率高時，請使用以下工具：

- Ethalyzer — 監控進出該CPU的流量型別。
- Configuration — 檢查交換器/介面/功能組態
- CoPP/硬體速率限制器 — 確保CoPP和HWRL配置正確。有時CPU運行率不高，因為它受到CoPP和速率限制器的保護。檢查CoPP和HWRL，檢視某些流量/資料包是否存在丟包。



注意:CoPP和HWRL都只能從預設虛擬裝置環境(VDC)中獲得。它們由每個單獨的I/O模組實施。來自多個模組的聚合流量仍然會給CPU帶來沉重負擔。

高CPU使用率的根本原因分析

網路中斷可以通過使用者干預來解決，也可以自行恢復。如果您懷疑CPU使用率高導致了網路中斷，請使用以下原則調查原因。

症狀

CPU使用率高的症狀包括控制平面不穩定、由控制平面故障引起的資料平面連線問題、協定抖動(如熱待機路由器協定(HSRP)/RP抖動)、UDLD錯誤禁用、生成樹協定(STP)故障以及其他連線問題。

CPU歷史記錄

show processes cpu history命令

如果交換機未重新載入或切換，請在中斷後72小時內運行show processes cpu history命令，以檢視事件發生時是否出現CPU使用率較高的情況。

CoPP和HWRL

如果CPU使用率高是過去中斷的根本原因，並且您懷疑中斷是由網路流量觸發的，可以使用CoPP和HWRL（硬體速率限制器）來幫助識別流量型別。

show policy-map interface control-plane命令

以下是show policy-map interface control-plane命令的輸出示例：

```
switch# show policy-map interface control-plane
Control Plane

  service-policy input: copp-system-p-policy-strict

  class-map copp-system-p-class-critical (match-any)
    match access-group name copp-system-p-acl-bgp
    match access-group name copp-system-p-acl-bgp6
    match access-group name copp-system-p-acl-igmp
    match access-group name copp-system-p-acl-msdp
    match access-group name copp-system-p-acl-ospf

    match access-group name copp-system-p-acl-pim
    match access-group name copp-system-p-acl-pim6
    match access-group name copp-system-p-acl-rip
    match access-group name copp-system-p-acl-rip6
    match access-group name copp-system-p-acl-vpc
    match access-group name copp-system-p-acl-eigrp
    match access-group name copp-system-p-acl-eigrp6
    match access-group name copp-system-p-acl-mac-l2pt
    match access-group name copp-system-p-acl-mps-ldp
    match access-group name copp-system-p-acl-mps-oam
    match access-group name copp-system-p-acl-ospf6
    match access-group name copp-system-p-acl-otv-as
    match access-group name copp-system-p-acl-mac-otv-isis
    match access-group name copp-system-p-acl-mps-rsvp
    match access-group name copp-system-p-acl-mac-fabricpath-isis
    match protocol mpls router-alert
    match protocol mpls exp 6
    set cos 7
    police cir 39600 kbps , bc 250 ms
    module 1 :
      conformed 1108497274 bytes; action: transmit
      violated 0 bytes; action: drop

    module 3 :
      conformed 0 bytes; action: transmit
      violated 0 bytes; action: drop

    module 10 :
      conformed 0 bytes; action: transmit
  .
  .
  .
```

show hardware rate-limiter mod <x>命令

以下是NX-OS版本6.1之前的show hardware rate-limiter mod 1命令的输出示例：

```
switch# show hardware rate-limiter mod 1

Units for Config: packets per second
Allowed, Dropped & Total: aggregated since last clear counters

Rate Limiter Class                Parameters
-----
layer-3 mtu                       Config      : 500
                                   Allowed       : 0
                                   Dropped      : 0
                                   Total          : 0

layer-3 ttl                       Config      : 500
                                   Allowed       : 0
                                   Dropped      : 0
                                   Total          : 0

layer-3 control                   Config      : 10000
                                   Allowed       : 0
                                   Dropped      : 0

.
.
.
```

以下是NX-OS 6.1版或更高版本中的show hardware rate-limiter mod 1命令的输出示例：

```
switch# show hardware rate-limiter mod 1
switch# show hardware rate-limiter module 1

Units for Config: packets per second
Allowed, Dropped & Total: aggregated since last clear counters

Module: 1
R-L Class           Config      Allowed     Dropped     Total
-----
L3 mtu              500         0           0           0
L3 ttl              500         0           0           0
L3 control          10000      0           0           0
L3 glean            100         0           0           0
L3 mcast dirconn   3000        0           0           0
L3 mcast loc-grp   3000        0           0           0
L3 mcast rpf-leak  500         0           0           0
L2 storm-ctrl      Disable
access-list-log     100         0           0           0
copy                30000      0           0           0
receive             30000      40583      0           40583
L2 port-sec         500        20435006   0           20435006
L2 mcast-snoop     10000      0           0           0
L2 vpc-low          4000        0           0           0
L2 l2pt             500         0           0           0
f1 r1-1            4500        0           0
f1 r1-2            1000        0           0
f1 r1-3            1000        0           0
```

```

f1 r1-4          100          0
f1 r1-5          1500         0
L2 vpc-peer-gw  5000         0          0
L2 lisp-map-cache 5000         0          0

```

查詢丟棄計數遞增的任何類。找出超過所配置閾值的類是否正常。

帶內驅動程式

show hardware internal cpu-mac inband [counters | 統計資訊 | events]命令

使用以下命令檢查CPU路徑中的丟棄、XOFF流量控制、最大CPU接收和傳輸速率等。

```

switch# show hardware internal cpu-mac inband stats
i82571 registers
=====
RMON counters                Rx                Tx
-----+-----+-----
total packets                70563313         139905960
good packets                 70563313         139905960
64 bytes packets              0                 0
65-127 bytes packets         66052368         135828505
128-255 bytes packets        1424632          1327796
256-511 bytes packets        280422           325220
512-1023 bytes packets       17060            14480
1024-max bytes packets       2788831         2409959

broadcast packets            0                 0
multicast packets            0                 0
good octets (hi)              0                 0
good octets (low)            18573099828      25929913975
total octets (hi)             0                 0
total octets (low)            18573090123      25929922452
XON packets                   0                 0
XOFF packets                   0                 0
-----> Pause Frame back to R2D2 when the traffic exceeds SUP limit
management packets           0                 0

Interrupt counters
-----+-----
Mine                          57079706
Other                          0
Assertions                     57079706
Rx packet timer                 9638
Rx absolute timer               0
Rx overrun                      0
Rx descr min thresh            0
Tx packet timer                 4189
Tx absolute timer               6476
Tx queue empty                  0
Tx descr thresh low            0
txdw ..... 44983549
txqe ..... 2
lsc ..... 0

```

rxseq 0
rxdmt 213229
rxo 0
rxt 32433891
mdac 0
rxcfg 0
gpi 0

Error counters

-----+-----
CRC errors 0
Alignment errors 0
Symbol errors 0
Sequence errors 0
RX errors 0
Missed packets (FIFO overflow) 0
Single collisions 0
Excessive collisions 0
Multiple collisions 0
Late collisions 0
Collisions 0
Defers 0
Tx no CRS 0
Carrier extension errors 0

Rx length errors 0
FC Rx unsupported 0
Rx no buffers 0 ----- no buffer
Rx undersize 0
Rx fragments 0
Rx oversize 0
Rx jabbers 0
Rx management packets dropped .. 0
Tx TCP segmentation context 0
Tx TCP segmentation context fail 0

Throttle statistics

-----+-----
Throttle interval 2 * 100ms
Packet rate limit 32000 pps
Rate limit reached counter .. 0
Tick counter 2132276
Active 0
Rx packet rate (current/max) 169 / 610 pps ----- Rx rate (current/max)
Tx packet rate (current/max) 429 / 926 pps

NAPI statistics

-----+-----
Weight 64
Poll scheduled . 57079706
Poll rescheduled 0
Poll invoked ... 117135124
Weight reached . 9
Tx packets 139905960
Rx packets 70563313
Rx congested ... 0
Rx redelivered . 0

qdisc stats:

-----+-----
Tx queue depth . 1000
qlen 0

```

packets ..... 139905960
bytes ..... 23411617016
drops ..... 0

```

Bahrain registers (cleared by chip reset only)

```

=====
revision          0x00000108
scratchpad        0xaaaaaaaa
MAC status        0x00000001
MAC SerDes synced 0x00000001
MAC status 2      0x000100f8
Auto-XOFF config  1
Auto-XOFF status  0

```

| MAC counters | MAC0 (R2D2) | | MAC1 (CPU) | |
|-----------------------------|-------------|------------|------------|------------|
| | Rx | Tx | Rx | Tx |
| 64 bytes packets | 0 | 0 | 0 | 0 |
| 65-127 bytes packets | 66907289 | 136682635 | 135828505 | 66052368 |
| 128-255 bytes packets | 570131 | 473705 | 1327796 | 1424632 |
| 256-511 bytes packets | 280003 | 325182 | 325220 | 280422 |
| 512-1023 bytes packets | 17061 | 14482 | 14480 | 17060 |
| 1024-1518 bytes packets | 623614 | 242009 | 241831 | 623569 |
| 1519-max bytes packets | 2165215 | 2167947 | 2168128 | 2165262 |
| total packets | 70563313 | 139905960 | 139905960 | 70563313 |
| total bytes | 405350248 | 2496404376 | 160120520 | 1393236630 |
| undersized packets | 0 | | 0 | |
| fragmented packets | 0 | | 0 | |
| FCS errors | 0 | | 0 | |
| auto-XOFF state entered | 0 times | | | |
| auto-XOFF reset | 0 times | | | |
| XOFF packets auto-generated | | 0 | | |
| XOFF packets | | 0 | 0 | |
| XON packets | 0 | | 0 | |
| parity error | 0 | 0 | 0 | 0 |
| fifo errors | 0 | | 0 | |
| overflow errors | | 0 | | 0 |

在NX-OS版本5.X之後，「events」是一個命令選項，用於提供達到每秒最大資料包(PPS)接收(RX)或傳輸(TX)CPU速率的時間。此範例顯示如何確定遇到最後一個CPU流量高峰的時間：

```
switch# show hardware internal cpu-mac inband events
```

- 1) Event:TX_PPS_MAX, length:4, at 648617 usecs after Fri Oct 19 13:23:06 2012
new maximum = 926
- 2) Event:TX_PPS_MAX, length:4, at 648622 usecs after Fri Oct 19 13:15:06 2012
new maximum = 916

- 3) Event:TX_PPS_MAX, length:4, at 648612 usecs after Fri Oct 19 13:14:06 2012
new maximum = 915
- 4) Event:TX_PPS_MAX, length:4, at 648625 usecs after Fri Oct 19 13:12:06 2012
new maximum = 914
- 5) Event:TX_PPS_MAX, length:4, at 648626 usecs after Fri Oct 19 13:11:06 2012
new maximum = 911
- 6) Event:TX_PPS_MAX, length:4, at 648620 usecs after Fri Oct 19 13:08:06 2012
new maximum = 910

show system internal pktmgr internal vdc inband <int>命令

使用此命令可以確定傳送到CPU的流量的來源。

```
switch# show system internal pktmgr internal vdc inband e1/5
Interface          Src Index      VDC ID      Packet rcvd
-----
Ethernet1/5        0xa1d         1           14640
```

Netstack/Pktmgr

Netstack是在Nexus 7000的使用者空間中實施的完整IP堆疊。元件包括L2資料包管理器、ARP、鄰接管理器、IPv4、網際網路控制消息協定v4(ICMPv4)、IPv6、ICMPv6、TCP/UDP和套接字型檔。當發往CPU的流量觸發CPU使用率較高時，您經常會看到Netstack及其各自的進程運行較高。

show system inband queuing status命令

此範例顯示如何顯示正在使用的Netstack佇列演演算法：

```
switch# show system inband queuing status
Weighted Round Robin Algorithm
Weights BPDU - 32, Q0 - 8, Q1 - 4, Q2 - 2 Q3 - 64
```

show system inband queuing statistics命令

此示例顯示可核心載入模組(KLM)和使用空間進程中的計數器。

KLM是在預設VDC上運行並在帶內和管理介面上運行的單個例項。KLM僅在入口資料包處理期間進入圖片，以便將入口幀傳送到右側VDC Netstack進行處理。

```
switch# show system inband queuing statistics
  Inband packets unmapped to a queue: 0
  Inband packets mapped to bpdu queue: 7732593
  Inband packets mapped to q0: 686667
  Inband packets mapped to q1: 0
  Inband packets mapped to q2: 0
  Inband packets mapped to q3: 20128
  In KLM packets mapped to bpdu: 7732593
  In KLM packets mapped to arp : 912
  In KLM packets mapped to q0 : 686667
  In KLM packets mapped to q1 : 0
  In KLM packets mapped to q2 : 0
  In KLM packets mapped to q3 : 20128
  In KLM packets mapped to veobc : 0
  Inband Queues:
  bpdu: rcv 1554390, drop 0, congested 0 rcvbuf 2097152, sndbuf 262142 no drop 1
  (q0): rcv 686667, drop 0, congested 0 rcvbuf 2097152, sndbuf 262142 no drop 0
  (q1): rcv 0, drop 0, congested 0 rcvbuf 2097152, sndbuf 262142 no drop 0
  (q2): rcv 0, drop 0, congested 0 rcvbuf 2097152, sndbuf 262142 no drop 0
  (q3): rcv 20128, drop 0, congested 0 rcvbuf 2097152, sndbuf 262142 no drop 0
```

show system internal pktmgr internal vdc global-stats命令

此命令類似於前面的show system inband queuing statistics命令，提供了許多詳細資訊：

```
switch# show system internal pktmgr internal vdc global-stats
```

```
VDC KLM global statistics:
  Inband packets not mapped to a VDC: 0
  Inband diag packets received: 998222
  Weighted Round Robin Algorithm
  Weights BPDU - 32, Q0 - 8, Q1 - 4, Q2 - 2 Q3 - 64
  Inband packets unmapped to a queue: 0
  Inband packets mapped to bpdu queue: 7734430 (7734430)
  Inband packets mapped to q0: 686779 (686779)
  Inband packets mapped to q1: 0 (0)
  Inband packets mapped to q2: 0 (0)
  Inband packets mapped to q3: 20128 (20128)
  Pkt Size History : 2811395 for index 1
  Pkt Size History : 274508 for index 2
  Pkt Size History : 74284 for index 3
  Pkt Size History : 43401 for index 4
  Pkt Size History : 70915 for index 5
  Pkt Size History : 35602 for index 6
  Pkt Size History : 30085 for index 7
  Pkt Size History : 29408 for index 8
  Pkt Size History : 21221 for index 9
  Pkt Size History : 15683 for index 10
  Pkt Size History : 13212 for index 11
  Pkt Size History : 10646 for index 12
  Pkt Size History : 9290 for index 13
  Pkt Size History : 50298 for index 14
  Pkt Size History : 5473 for index 15
  Pkt Size History : 4871 for index 16
  Pkt Size History : 4687 for index 17
```

```
Pkt Size History : 5507 for index 18
Pkt Size History : 15416 for index 19
Pkt Size History : 11333 for index 20
Pkt Size History : 5478 for index 21
Pkt Size History : 4281 for index 22
Pkt Size History : 3543 for index 23
Pkt Size History : 3059 for index 24
Pkt Size History : 2228 for index 25
Pkt Size History : 4390 for index 26
Pkt Size History : 19892 for index 27
Pkt Size History : 524 for index 28
Pkt Size History : 478 for index 29
Pkt Size History : 348 for index 30
Pkt Size History : 447 for index 31
Pkt Size History : 1545 for index 32
Pkt Size History : 152 for index 33
Pkt Size History : 105 for index 34
Pkt Size History : 1424 for index 35
Pkt Size History : 43 for index 36
Pkt Size History : 60 for index 37
Pkt Size History : 60 for index 38
Pkt Size History : 46 for index 39
Pkt Size History : 58 for index 40
Pkt Size History : 829 for index 41
Pkt Size History : 32 for index 42
Pkt Size History : 26 for index 43
Pkt Size History : 1965 for index 44
Pkt Size History : 21 for index 45
Pkt Size History : 1 for index 46
Pkt Size History : 1 for index 48
Pkt Size History : 1 for index 51
Pkt Size History : 1 for index 52
Pkt Size History : 1 for index 53
Pkt Size History : 3 for index 55
In KLM packets mapped to bpdu: 7734430
In KLM packets mapped to arp : 912
In KLM packets mapped to q0 : 686779
In KLM packets mapped to q1 : 0
In KLM packets mapped to q2 : 0
In KLM packets mapped to q3 : 20128
In KLM packets mapped to veobc : 0
In KLM Queue Mapping (0 1 2 3 4)
Data Available in FDs (0 0 0 0 0)
Inband Queues:
bpdu: rcv 1556227, drop 0, congested 0 rcvbuf 2097152, sndbuf 262142 no drop 1
(q0): rcv 686779, drop 0, congested 0 rcvbuf 2097152, sndbuf 262142 no drop 0
(q1): rcv 0, drop 0, congested 0 rcvbuf 2097152, sndbuf 262142 no drop 0
(q2): rcv 0, drop 0, congested 0 rcvbuf 2097152, sndbuf 262142 no drop 0
(q3): rcv 20128, drop 0, congested 0 rcvbuf 2097152, sndbuf 262142 no drop 0
Mgmt packets not mapped to a VDC: 227551
Mgmt multicast packets dropped: 92365
Mgmt multicast packets delivered: 0
Mgmt packets broadcast to each VDC: 23119
Mgmt debugging packets copied: 0
Mgmt IPv6 multicast packets delivered: 0
Mgmt IPv6 link-local packets delivered: 0
Mgmt LLDP packets received: 0
```

show system internal pktmgr interface ethernet <int>命令

使用以下命令檢視來自介面的CPU傳出流量的封包速率和流量型別 (單點傳送或多點傳送) 。

```
switch# show system internal pktmgr interface e1/5
Ethernet1/5, ordinal: 73
SUP-traffic statistics: (sent/received)
  Packets: 63503 / 61491
  Bytes: 6571717 / 5840641
  Instant packet rate: 0 pps / 0 pps
  Packet rate limiter (Out/In): 0 pps / 0 pps
  Average packet rates(1min/5min/15min/EWMA):
  Packet statistics:
    Tx: Unicast 3198, Multicast 60302
       Broadcast 3
    Rx: Unicast 3195, Multicast 58294
       Broadcast 2
```

show system internal pktmgr client <uuid>命令

此命令顯示在Packet Manager中註冊的應用(例如STP或Cisco Discovery Protocol(CDP))以及這些應用傳送和接收的資料包數。

```
switch# show system internal pktmgr client
Client uuid: 268, 4 filters, pid 3127
  Filter 1: EthType 0x0806,
  Rx: 2650, Drop: 0
  Filter 2: EthType 0xffff0, Exc 8,
  Rx: 0, Drop: 0
  Filter 3: EthType 0x8841, Snap 34881,
  Rx: 0, Drop: 0
  Filter 4: EthType 0x0800, DstIf 0x150b0000, Excl. Any
  Rx: 0, Drop: 0
Options: TO 0, Flags 0x18040, AppId 0, Epid 0
Ctrl SAP: 278, Data SAP 337 (1)
Total Rx: 2650, Drop: 0, Tx: 1669, Drop: 0
Recirc Rx: 0, Drop: 0
Rx pps Inst/Max: 0/20
Tx pps Inst/Max: 0/5
COS=0 Rx: 0, Tx: 0    COS=1 Rx: 912, Tx: 0
COS=2 Rx: 0, Tx: 0    COS=3 Rx: 0, Tx: 0
COS=4 Rx: 0, Tx: 0    COS=5 Rx: 0, Tx: 1669
COS=6 Rx: 0, Tx: 0    COS=7 Rx: 1738, Tx: 0
```

```
Client uuid: 270, 1 filters, pid 3128
  Filter 1: EthType 0x86dd, DstIf 0x150b0000, Excl. Any
  Rx: 0, Drop: 0
Options: TO 0, Flags 0x18040, AppId 0, Epid 0
Ctrl SAP: 281, Data SAP 283 (1)
Total Rx: 0, Drop: 0, Tx: 0, Drop: 0
Recirc Rx: 0, Drop: 0
Rx pps Inst/Max: 0/0
Tx pps Inst/Max: 0/0
COS=0 Rx: 0, Tx: 0    COS=1 Rx: 0, Tx: 0
COS=2 Rx: 0, Tx: 0    COS=3 Rx: 0, Tx: 0
COS=4 Rx: 0, Tx: 0    COS=5 Rx: 0, Tx: 0
```

COS=6 Rx: 0, Tx: 0 COS=7 Rx: 0, Tx: 0

show system internal pktmgr stats命令

使用以下命令以檢查封包是否在輸入路徑中到達封包管理員，以及封包管理員是否正在將封包送出。此命令還可以幫助您確定接收或傳輸路徑中的緩衝區是否存在問題。

```
switch# show system internal pktmgr stats
Route Processor Layer-2 frame statistics
```

```
Inband driver: valid 1, state 0, rd-thr 1, wr-thr 0, Q-count 0
Inband sent: 56441521, copy_drop: 0, ioctl_drop: 0,
  unavailable_buffer_hdr_drop: 0
Inband standby_sent: 0
Inband encap_drop: 0, linecard_down_drop: 0
Inband sent by priority [0=11345585,5=164281,6=43280117,7=1651538]
Inband max output queue depth 0
Inband recv: 89226232, copy_drop: 0, ioctl_drop: 0,
  unavailable_buffer_hdr_drop: 0
Inband decap_drop: 0, crc_drop: 0, recv by priority: [0=89226232]
Inband bad_si 0, bad_if 0, if_down 0
Inband last_bad_si 0, last_bad_if 0, bad_di 0
Inband kernel recv 44438488, drop 0, rcvbuf 2097152, sndbuf 4194304
```

```
Mgmt driver: valid 1, state 0, rd-thr 1, wr-thr 0, Q-count 0
Mgmt sent: 971834, copy_drop: 0, ioctl_drop: 0,
  unavailable_buffer_hdr_drop: 0
Mgmt standby_sent: 0
Mgmt encap_drop: 0, linecard_down_drop: 0
Mgmt sent by priority [0=925871,5=45963]
Mgmt max output queue depth 0
Mgmt recv: 1300932, copy_drop: 0, ioctl_drop: 0,
  unavailable_buffer_hdr_drop: 0
Mgmt decap_drop: 0, crc_drop: 0, recv by priority: [0=1300932]
Mgmt bad_si 0, bad_if 0, if_down 0
Mgmt last_bad_si 0, last_bad_if 0, bad_di 0
Mgmt kernel recv 1300932, drop 0, rcvbuf 2097152, sndbuf 2097152
```

```
Inband2 driver: valid 0, state 1, rd-thr 0, wr-thr 0, Q-count 0
```

```
No of packets passed by    PM Policy database            876452
No of packets dropped by   PM Policy database            0
No of packets bypassed by PM Policy database            424480
No of packets dropped by   PM originating from kernel 0
```

```
MBUF SK Tx: 57413355 pkts (requested 57413355 denied 0), 62236110 mbufs
  function invoked 57413355 denied 0/0 c/realloc 0/0
MBUF SK Rx: 90527161 pkts, 90527421 mbufs (requested 2388154951 denied 0)
  function invoked 35132836
```

```
Global input drops: bad-interface 0, bad-encap 0, failed-decap 0,
  no prot 42371
recv_encap_type_err 0, recv_decap_err 0,  recv_mac_mismatch 0, recv_no_client 0
recv_no_svi 0, recv_no_vlan 0,  recv_client_notreg 0, recv_enqueue_fail 0
```

```
Global output drops:
```

send_ifdown_fail 13, send_invalid_iid 0
send_invalid_vlan 0, send_security_drop 0 send_loopback_drop 0,
send_small_pkt_fail 0
send_vs1_err 0, send_dce_err 0, send_enqueue_fail 0, send_alloc_fail 0

DCE errors:

misc_err 0, lookup_err 0, encap_err 0, decap_err 0

Platform errors:

generic_encap_err 0, encap_err 0, decap_err 0
vlan_encap_err 0, vlan_decap_err 0

DC3HDR errors:

pkt_err 0, vlan_err 0, ifidx_err 0, portidx_err 0

RECIRC errors:

misc_err 0, lookup_err 0

Lcache errors:

init_err 0, timer_err 0

Stats errors:

misc_err 0, init_err 0, timer_err 0

Client errors:

alloc_err 0, pid_err 0, register_err 0, unregister_err 0
add_err 0, delete_err 0, update_err 0

VDC errors:

alloc_err 0, set_err 0, update_err 0

Misc. errors:

mts_err 0, mbuf_err 0, drop_exception 0
invalid_drv_type 0, interface_err 0
eth_output_err 0, gre_err 0 otv_err 0
tunnel_6to4_err 0, mcec_err 0, invalid_gpc 0 invalid_ftag 0 invalid_l2_type :0
register_err 0, unregister_err 0, invalid_args 0, file_open_err 0
inband_err 0, vlan_err 0, pm_alloc_err 0, pm_ha_err 0, pm_init_err 0
arp_init_err 0, rtm_init_err 0, am_init_err 0, ui_init_err 0, mpls_init_err 0,
evc_init_err 0
sdb_err 95670, sdb_init_err 0
sysmgr_err 0, eth_span_err 0, buf_pool_err 0, feature_err 0
uuid2client_err 16, dot1q_drop 0, nfcache_init_err 0

Crossbar down drops : 0

Exception packets: mtu-fail 0, icmp-redirect 0, icmp-unreach 0, ttl 0
options 0, rpf 0, two-mcast-rpf 0, l3-bridge-drop 0
mcast-next-hop 0, multicast 0
drop 0, acl-redirect 0, acl-redirect-arp 0, acl-redirect-dhcp 0
sup-shim-pkt 229385 Pkts recvd with peergway SUP DI 0

VPC Frame Statistics

VPC Mgr reg state 1, im-ext-sdb-state 1
Ingress BPDUs qualified for redirection 0
Ingress BPDUs redirected to peer 0
Egress BPDUs qualified for redirection 0
Egress BPDUs dropped due to remote down 0
Egress BPDUs redirected to peer 0
Ingress pkts qualified for peergateway tunneling 0
Ingress pkts tunneled to peer with peergateway conf 0
Peer-gw pkts tunneled tx :
From VPC+ leg 0, From VPC leg 0, From l2mp network 0

From orphan port in VPC+ 0, from orphan port in VPC 0
For ARP 0, IP 0, IPv6 0, unknown 0
Total Tunneled packets received from peer 0
Local delivery 0, Transmit down 0, peer-gw tunneled 0
Tunnel rx packets drop due to local vpc leg down 0
Peer-gw pkts tunneled rx :
From VPC+ leg 0, VPC leg 0, From l2mp network 0
From orphan port in VPC+ 0, from orphan port in VPC 0
For ARP 0, IP 0, IPv6 0, unknown 0

Error Statistics

VPC manager: uninit 0, library 0
Tunnel (ingress): non-mct rx 0, bad hdr 0, badpkts 0, non gpc peer 0
Tunnel (ingress): redirlooperror 0
Tunnel (egress): in-bpdu 0, e-bpdu 0, peer-gw 0
Mbuf: alloc: 0, prepend: 0, pullup: 0
Invalid filter: 0
Peergw tunneling tx: invalid ftag 0, invalid swid 0
invalid iftype 0, invalid GPC of peer 0
Peergw tunneling rx: invalid msg subtype 0, invalid GPC of core 0
invalid GPC of peer 0, invalid svi 0
Unicast pkts which passed egress redirection check 0

statistics last reset 2w0d

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）。