排除故障在ASR1000系列路由器的高CPU

目录

简介

本文描述如何排除故障在一个ASR1000系列路由器的高CPU问题。

前提条件

要求

思科建议您了解ASR1000体系结构解释和使用本文。

说明

在Cisco路由器的高CPU可能定义作为在路由器的CPU利用率在正常使用情况上的情况。在某些情况下，当可能在其他情况下指示问题时，增加的CPU使用情况预计。在路由器的瞬变高CPU利用率由于网络更改或配置更改可以忽略并且是预料之中的行为。

然而，体验高CPU利用率在延长期限，不用在网络或配置上的任何变化的路由器是异常的并且需要被分析。 所以，当过度使用， CPU不能积极地服务其他进程，导致缓慢的line命令、控制层面延迟、丢包和服务的失败。

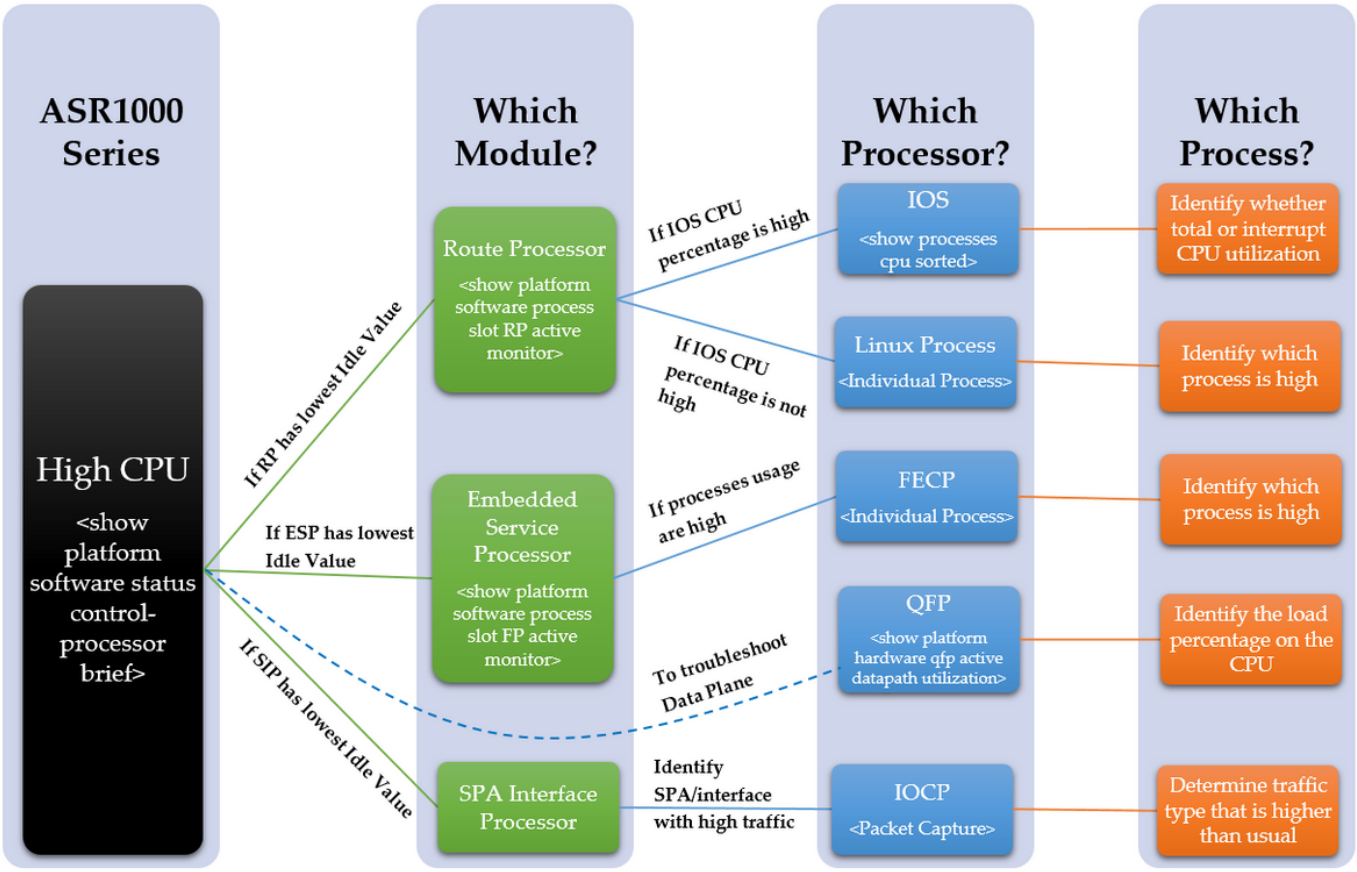
高CPU的原因是：

1. 控制层面CPU接收许多个平底船流量
2. 意外地正常运行并且导致CPU过度使用的进程

3. 数据层面处理器是被过度利用的/oversubscribed
4. 许多处理器中断

因为路由器CPU利用率是正比例的对在路由器的负载高CPU总是不是一ASR1000系列路由器。 例如，如果有网络更改，这将导致很多控制层面流量，因为网络将再聚合。所以，我们需要确定CPU过度使用的根本原因确定它是否是预料之中的行为或问题。

下面选派关于怎样的一逐步进程排除故障高CPU问题的图表：



排除故障步骤

Step1 –识别有高CPU的模块

ASR1000有在不同的模块间的几个不同的CPU。所以，我们需要发现哪个模块显示非常地比正常使用情况。这能通过空闲值被看到，更低空闲值，越高该模块CPU利用率。这些不同的CPU全部反射模块的控制层面。

确定在设备内的哪个模块被观察体验高CPU。是它RP、ESP或者SIP用下面的命令

参考下面的输出查看选中项目列

如果RP有低值空闲，则请继续到步骤2点1

如果ESP有低值空闲，则请继续到步骤3点2

如果SIP有低空闲值，则请继续到步骤4点3

```
Router#show


Slot1 MIN 5 MIN 15 MIN
RP00.00 0.02 0.00
ESP00.01 0.02 0.00
SIP00.00 0.01 0.00

(kb)
Slot(Pct) (Pct)(Pct)
RP02009376 1879196 (94%) 130180 (6%) 1432748 (71%)
ESP02009400 692100 (34%) 1317300 (66%) 472536 (24%)
SIP0471804 284424 (60%) 187380 (40%) 193148 (41%)

CPU
Slot CPUIRQ SIRQ IOwait
RP0 0 2.59 2.49 0.00 94.80 0.00 0.09 0.00
ESP0 0 2.30 17.90 0.00 79.80 0.00 0.00 0.00
SIP0 0 1.29 4.19 0.00 94.41 0.09 0.00 0.00
```

如果空闲值所有相对高，它可能不是控制层面问题。排除故障ESP的QFP需要被观察的数据层面。“高CPU”症状可以仍然被观察的归结于被过度利用的QFP，不会导致在控制层面处理器的高CPU。继续执行步骤 6。

步骤2 –分析模块

- **路由处理器**

在处理器被观察有高CPU利用率用下面的命令的RP内的确认。它是否是Linux进程或IOS ？

```
slot RP
```

如果IOS CPU百分比高(linux_iosd-imag)，则是RP IOS。继续对STEP 3

如果其他进程的CPU百分比高，则是Linux进程可能的。继续对STEP 4

- **嵌入式服务处理器**

在ESP内的确认，如果控制层面处理器被观察有高CPU利用率。它是否是FECP ？

```
slot FP
```

如果进程高那么它是FECP，则继续对STEP 5

如果它不是FECP，它不是在ESP内的控制层面进程相关问题。如果症状例如网络延迟或队列丢弃仍然被观察，数据层面可能需要为过度使用查看。继续对STEP 6

- **SPA接口处理器**

如果SIP被观察安排高CPU利用率然后IOCP请被观察有高CPU。确定哪些进程或进程在IOCP内被观察有高CPU利用率。

执行一数据包捕获并且识别哪个流量高于通常，并且哪些进程关联与此种流量。 继续对STEP 7

步骤3 – IOS进程

参考下面的输出，第一百分比是总CPU利用率，并且第二百分比是中断CPU利用率，是CPU用于的相当数量处理被踢的数据包。

如果中断百分比高那么表示很多流量被踢对RP， (此可以确认与show platform命令**软件结构平底船**)

如果中断百分比低，但是总计CPU高，则有将被观察作为扩展周期使用CPU的进程或进程。

在处理的IOS内的确认或进程被观察有高CPU利用率用下面的命令。

```
show processes cpu
```

识别哪百分比高(总CPU或中断CPU)，如果必须然后识别单个进程/进程。继续对STEP 7

```
Router#show cpu

CPU0%/0%;1%;1%

PID Runtime(ms)uSecs 5sec 1Min 5Min TTY

PID Runtime(ms)uSecs 5sec 1Min 5Min TTY

188 8143 434758 18 0.15% 0.18% 0.19% 0Ti

515 380 7050 53 0.07% 0.00% 0.00% 0 SBC

 3 2154 215 10018 0.07% 0.00% 0.19% 0 Exec

380 1783 55002 32 0.07% 0.06% 0.06% 0MMA DB

63 3132 11143 281 0.07% 0.07% 0.07% 0 IOSD ipc

 5 1 2 500 0.00% 0.00% 0.00% 0IPC ISSU Dispatc

 6 19 12 1583 0.00% 0.00% 0.00% 0RF

 8 0 1 0 0.00% 0.00% 0.00% 0RO

 7 0 1 0 0.00% 0.00% 0.00% 0EDDRI_MAIN

10 6 75 80 0.00% 0.00% 0.00% 0

 9 5671 538 10540 0.00% 0.14% 0.12% 0
```

步骤4 – Linux进程

如果IOS被观察过度了利用CPU，则我们需要对单个Linux进程观察CPU利用率。这些进程是从**显示平台软件进程slot RP激活监控器**列出的其他进程。识别处理或进程被观察体验高CPU然后继续对STEP 7。

步骤5 – FECP进程

如果进程或进程高然后它是可能的那些是在对高CPU利用率负责的FECP内的进程，请继续对STEP 7

步骤6 – QFP利用率

Quantum流处理器是转发ASIC。要确定在转发引擎的负载， QFP可以是受监视。下面的命令一览表输入和输出数据包(优先级和非优先)在数据包每秒和比特/秒。最终线路显示总量CPU负载由于在百分比的信息包转发。

```
qfp
```

请识别，如果输入或输出高，并且查看加工量然后继续对STEP 7

```
Router#show qfp
 CPP 0 Subdev 0 51560
  (pps) 0 0 0 0
        (/) 208 176 176 176
   (pps) 0 2 2 2
        (/) 64 784 784 784
     (pps) 0 2 2 2
       (/) 272 960 960 960
 (pps) 0 0 0 0
        (/) 192 160 160 160
   (pps) 0 1 1 1
        (/) 0 6488 6496 6488
     (pps) 0 1 1 1
        (/) 192 6648 6656 6648
(pct) 0 0 0 0
```

步骤7 –确定根本原因并且识别修正

使用被观察过度了利用识别的CPU，有一张更加清楚的图片的进程或进程高CPU为什么发生。要继续，请研究已确定进程执行的功能。这将协助解决确定关于怎样的一行动方案看待问题。例如-如果进程对特定协议负责那么您可以要查看与此协议涉及的配置。

如果仍然遇到CPU相关问题，推荐与TAC联系允许工程师帮助您进一步排除故障。排除故障的上述步骤将帮助工程师效率更高查出问题。

排除故障示例

在本例中我们通过进程将运作排除故障，并且尝试到最佳请识别路由器的高CPU的一个可能的根本原因。要开始，请确定哪个模块被观察体验高CPU，我们有下面的输出：

```
Router#show

Slot1 MIN 5 MIN 15 MIN
RP00.66 0.15 0.05
ESP00.00 0.00 0.00
SIP00.00 0.00 0.00

(kb)
Slot(Pct) (Pct)(Pct)
RP02009376 1879196 (94%) 130180 (6%) 1432756 (71%)
ESP02009400 692472 (34%) 1316928 (66%) 472668 (24%)
SIP0471804 284556 (60%) 187248 (40%) 193148 (41%)

CPU
Slot CPUIRQ SIRQ IOwait
RP0 0 57.11 14.42 0.00 0.00 28.25 0.19 0.00
ESP0 0 2.10 17.91 0.00 79.97 0.00 0.00 0.00
SIP0 0 1.20 6.00 0.00 92.80 0.00 0.00 0.00
```

因为在RP0内的空闲数量非常低，建议在路由处理器内的高CPU问题。所以进一步排除故障我们将识别在RP内的哪个处理器被观察体验高CPU。

```
Router#showcpu
CPU84%/36%;34%;9%
PID Runtime(ms)uSecs 5sec 1Min 5Min TTY
107 303230 50749 5975 46.69% 18.12% 4.45% 0IOSXE-RPSe
63 105617 540091 195 0.23% 0.10% 0.08% 0 IOSD ipc
159 74792 2645991 28 0.15% 0.06% 0.06% 0VRRS
116 53685 169683 316 0.15% 0.05% 0.01% 0
 9 305547 26511 11525 0.15% 0.28% 0.16% 0
188 362507 20979154 17 0.15% 0.15% 0.19% 0Ti
 3 147 186 790 0.07% 0.08% 0.02% 0 Exec
 2 32126 33935 946 0.07% 0.03% 0.00% 0
446 416 33932 12 0.07% 0.00% 0.00% 0 VDC
164 59945 5261819 11 0.07% 0.04% 0.02% 0 IP ARP
43 1703 16969 100 0.07% 0.00% 0.00% 0IPCM
```

从此输出，可以注意到总计CPU百分比和中断百分比高于预期。使用CPU的顶部进程是进程处理RP CPU的流量的“IOSXE-RP平底船Se”，因此我们能调查进一步此流量被踢对RP。

```
Router#show
LSMPIstats
```

```
enabled=0 disabled=0 throttled=0 unthrottled=0
= 90100722
= 100439
rxdone count= 90100722
txdone count= 100436
Rxparticletype count= 0
Txparticletype count= 0
Shadow count= 0Txbuf
= 0
= 0
stats
Bad0
Bad0
0
0
0
0
Bad0
Bad0
Bad0
0
0
swidb 1
ESS0
ESS0
SSLVPN0
0
0
IOSXE-RP
  62210226Layer2
    147 ARP
  27801234
   84426 RP<->QFP keepalive
     6
    1647

  FOR_USIPv4 protcol stats
     1647OSPF
histogram(500/bin)92avg 56
PAK
   0+   90097805 98790
  500+       0 7
```

从此输出，我们能看到有在指示流量被处理往路由器的很多数据包“为我们数据包”，此计数器被确认是从命令的观察多次增加了在几分钟。这确认很多被踢的流量过度利用CPU，经常是控制层面流量。控制层面流量能包括ARP、SSH、SNMP，路由更新(BGP， EIGRP， OSPF)等。从此信息，我们能识别高CPU的潜在原因，并且这协助解决为根本原因排除故障。例如，数据包捕获或另外流量监视器可能实现发现确切的流量被踢对将允许根本原因识别和解决在将来防止一个相似的问题的RP。

一旦数据包捕获完成，可能性被踢的流量某些示例是：

- **ARP** ：这可能归结于过量的ARP请求，将发生，如果多个IP地址是发送ARP请求通过Ip

route的配置对广播接口。这可能也归结于从ARP表的被冲洗的条目，并且必须重学根据老化的MAC地址项，或者建立接口出现的/下来。

- **SSH** ：这可能导致高CPU由于一大show命令(show tech-support)或，当很多调试指令启用时，强制很多CLI在SSH会话发送。

- **SNMP** ：这可能归结于需要长时间时刻处理请求的SNMP代理程序，并且导致高CPU。经常两个可能原因是轮询的MIB，或者路由并且/或者由NMS轮询的ARP表。

- **路由更新**：通常路由更新汇集归结于网络再收敛，或者请连接飘荡。这可能指示在网络内去下来的路由，或者沿着走的整个设备哪些强制聚合和重新计算最佳路由的网络，取决于哪个路由协议是在使用中的。

这突出显示根本原因如何可以通过高CPU的原因的识别隔离，当下来到单个进程级别时。从这里，单个进程或协议在隔离可以被分析识别它是否是配置问题、软件问题、网络设计或者打算的实践。

其它命令

下面是其他另外的有用的命令列表使用和排序由哪些处理器他们关连与：

路由处理器

- **<show进程cpu history>** 在最后60秒、分钟和72个小时提供CPU历史记录的图表
- **<show进程*process_ID* >** 关于单个进程内存和CPU分配的详细信息
- **<show平台软件基础设施punt>** 在被踢对RP的所有流量提供信息
- **<show平台软件状态控制处理器brief>** 选派CPU的负载和“健康的，以及选派内存和模块统计信息
- **<show平台软件进程slot r0|r1监视器>** 选派不同的进程和他们的CPU和存储器分配在所选的模块
- **<monitor平台软件进程r0|r1>** 提供进程的更新的一个实际源，他们使用CPU在全局配置模式要求命令“terminal terminal-type”首先被输入正确地作用

嵌入式服务处理器

- **<show平台软件进程列表fp激活summary>** 选派在CPU运行所有进程，以及平均负载的摘要
- **<show平台软件进程slot f0|f1监视器>** 选派不同的进程和他们的CPU和存储器分配在所选的模块
- **<monitor平台软件进程f0|f1>** 提供他们使用CPU进程的更新的一个实际源，当他们是在全局配置模式要求命令“terminal terminal-type”首先被输入正确地作用