

排除故障在ASR1000系列路由器的高CPU

目录

[简介](#)

[前提条件](#)

[要求](#)

[说明](#)

[排除故障步骤](#)

[Step1 –识别有高CPU的模块](#)

[步骤2 –分析模块](#)

[步骤3 – IOS进程](#)

[步骤4 – Linux进程](#)

[步骤5 – FECF进程](#)

[步骤6 – QFP使用率](#)

[步骤7 –确定根本原因并且识别修正](#)

[排除故障示例](#)

[其它命令](#)

[路由处理器](#)

[嵌入式服务处理器](#)

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

思科建议您有更好地了解[ASR1000体系结构](#)。

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和预计。

然而体验高CPU利用率在延长期限的路由器没有在网络或配置上的任何变化是异常的并且需要被分析。此过度使用导致的CPU能积极地服务其他进程，造成缓慢的line命令、控制层面延迟、丢包和失败服务。

高CPU的原因是：

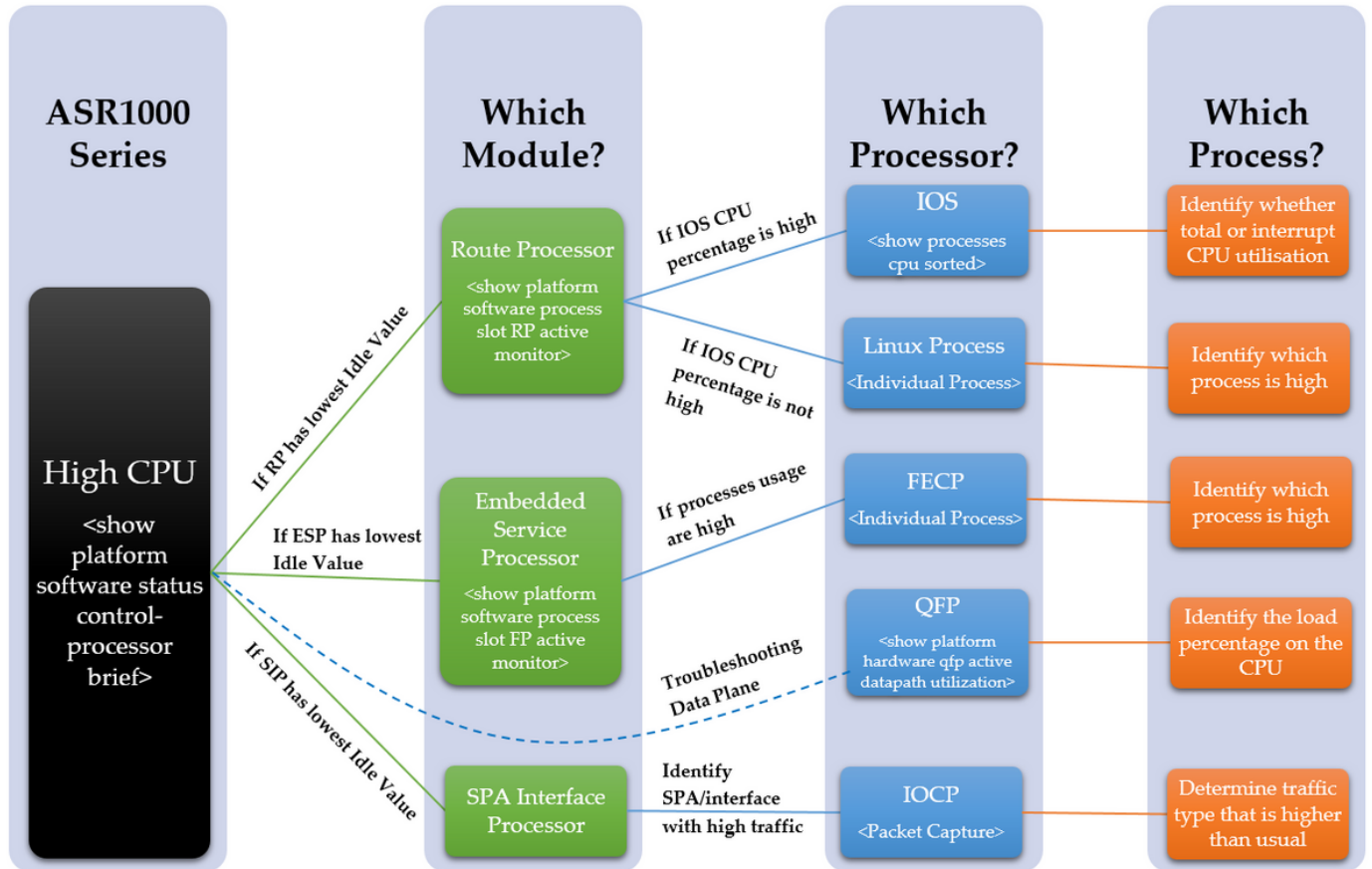
1. 控制层面CPU收到许多平底船流量
2. 拱起CPU的一行为不端的进程

3. 数据层面处理器在使用的/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，网络再聚合。所以，我们需要确定CPU的根本原因在利用率确定它是否是预料之中的行为或问题。

下面infographic选派关于怎样的一逐步进程排除故障高CPU问题：



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，越高该模块的CPU利用率。这些CPU全部反射模块的控制层面。

确定在设备内的哪个模块体验高CPU。是它RP、ESP或者SIP用下面的命令

显示平台软件状态控制处理器摘要

参考图1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平台软件状态控制处理器摘要

负载平均

Slot状态 1 MIN 5 MIN 15 MIN

RP0健康 0.00 0.02 0.00

ESP0健康 0.01 0.02 0.00

SIP0健康 0.00 0.01 0.00

内存(kb)

Slot状态 托塔尔使用(Pct) (Pct)任意做了(Pct)

RP0健康 2009376 1879196 (94%) 130180 (6%) 1432748 (71%)

ESP0健康 2009400 692100 (34%) 1317300 (66%) 472536 (24%)

SIP0健康 471804 284424 (60%) 187380 (40%) 193148 (41%)

CPU 利用率

Slot CPU用户系统 尼斯空闲 IRQ 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QFP在使用，不会导致报告高CPU的控制层面处理器。继续执行步骤 6。

步骤2 –分析模块

• 路由处理器

在使用下面的命令，处理器体验高CPU的RP内的确认。它是否是Linux进程或IOS？

显示平台软件进程slot RP激活监控器

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体验高CPU。它是否是FECP？

显示平台软件进程slot FP激活监控器

如果进程高那么它是FECP，则继续对STEP 5

如果它不是FECP它不是处理在ESP内的控制层面问题。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在使用率查看。继续对STEP 6

• SPA接口处理器

如果SIP体验高CPU那么IOCP体验高CPU。确定哪些进程或进程在IOCP内最使用的CPU

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

识别处理或进程最使用的CPU。继续对STEP 7

步骤3 – IOS进程

参考的图2，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，是处理被踢的数据包CPU使用的相当数量

如果中断百分比高那么表示很多流量被踢对RP，(这可以确认与显示平台软件基础设施平底船)

如果中断百分比低，但是总计CPU高，则有进程或处理作为扩展周期使用CPU对预计。

使用下面的命令，在处理的IOS内的确认或进程体验高CPU使用率。

排序的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处理排序的cpu
```

```
CPU利用率五秒：0%/0%;一分钟：1%;五分钟：1%
```

```
PID Runtime(ms)调用了uSecs 5sec 1Min 5Min TTY进程
```

```
PID Runtime(ms)调用了uSecs 5sec 1Min 5Min TTY进程
```

```
188 8143 434758 18 0.15% 0.18% 0.19% 0以太网毫秒Ti
```

```
515 380 7050 53 0.07% 0.00% 0.00% 0 SBC主要进程
```

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

```
380 1783 55002 32 0.07% 0.06% 0.06% 0个MMA DB计时器
```

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

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

```
6 19 12 1583 0.00% 0.00% 0.00% 0个RF从属美因河周四
```

```
8 0 1 0 0.00% 0.00% 0.00% 0个RO通知计时器
```

```
7 0 1 0 0.00% 0.00% 0.00% 0个EDDRI_MAIN
```

```
10 6 75 80 0.00% 0.00% 0.00% 0池管理器
```

```
9 5671 538 10540 0.00% 0.14% 0.12% 0检查堆栈
```

步骤4 – Linux进程

如果IOS不拱起CPU，则我们需要查看CPU利用率为单个Linux进程，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。识别处理或进程体验高CPU然后继续对STEP 7

步骤5 – FECF进程

如果进程或进程高然后它是可能的那些是在引起高CPU使用率的FEC P内的进程，请继续对STEP 7

步骤6 – QFP使用率

Quantum流处理器是转发ASIC。监控QFP将显示在转发引擎的负载。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。

显示平台硬件qfp活动数据路径利用率

请识别，如果输入或输出高，并且查看处理负荷然后继续对STEP 7

Router#show平台硬件qfp活动数据路径利用率

CPP 0 : Subdev 0 5秒1分钟5分钟60分钟

输入：优先级(pps) 0 0 0 0

(位/秒) 208 176 176 176

非优先(pps) 0 2 2 2

(位/秒) 64 784 784 784

托塔尔(pps) 0 2 2 2

(位/秒) 272 960 960 960

输出：优先级(pps) 0 0 0 0

(位/秒) 192 160 160 160

非优先(pps) 0 1 1 1

(位/秒) 0 6488 6496 6488

托塔尔(pps) 0 1 1 1

(位/秒) 192 6648 6656 6648

处理：负载(pct) 0 0 0 0

步骤7 –确定根本原因并且识别修正

使用拱起识别的CPU，有一张更加清楚的图片的进程或进程高CPU为什么处理器的发生或过使用率。研究已确定进程执行的功能。这将给我们关于怎样的行动方案看待问题。例如-如果进程对特定协议负责那么您可以要查看该配置。

如果仍然遇到问题，推荐与TAC联系允许工程师帮助您进一步排除故障。上述故障排除步骤将帮助工程师更加容易地查出问题。

排除故障示例

在本例中我们通过故障排除流程将运作，并且尝试到最佳请识别路由器高CPU的一个可能的根本原因。开始与识别哪个模块体验高CPU，我们有以下输出：

```
Router#show平台软件状态控制处理器摘要
```

```
负载平均
```

```
Slot状态1 MIN 5 MIN 15 MIN
```

```
RP0健康0.66 0.15 0.05
```

```
ESP0健康0.00 0.00 0.00
```

```
SIP0健康0.00 0.00 0.00
```

```
内存(kb)
```

```
Slot状态托塔尔使用(Pct) (Pct)任意做了(Pct)
```

```
RP0健康2009376 1879196 (94%) 130180 (6%) 1432756 (71%)
```

```
ESP0健康2009400 692472 (34%) 1316928 (66%) 472668 (24%)
```

```
SIP0健康471804 284556 (60%) 187248 (40%) 193148 (41%)
```

```
CPU 利用率
```

```
Slot CPU用户系统尼斯空闲IRQ 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，因此继续我们将识别的故障排除在RP内的哪个处理器体验高CPU。

```
Router#show处理排序的cpu
```

```
CPU利用率五秒：84%/36%；一分钟：34%；五分钟：9%
```

```
PID Runtime(ms)调用了uSecs 5sec 1Min 5Min TTY进程
```

```
107 303230 50749 5975 46.69% 18.12% 4.45% 0个IOSXE-RP平底船Se
```

```
63 105617 540091 195 0.23% 0.10% 0.08% 0 IOSD ipc任务
```

```
159 74792 2645991 28 0.15% 0.06% 0.06% 0个VRRS美因河线索
```

```
116 53685 169683 316 0.15% 0.05% 0.01% 0每第二乔布斯
```

```
9 305547 26511 11525 0.15% 0.28% 0.16% 0检查堆栈
```

```
188 362507 20979154 17 0.15% 0.15% 0.19% 0以太网毫秒Ti
```

```
3 147 186 790 0.07% 0.08% 0.02% 0 Exec
```

```
2 32126 33935 946 0.07% 0.03% 0.00% 0个负载计
```

```
446 416 33932 12 0.07% 0.00% 0.00% 0 VDC进程
```

```
164 59945 5261819 11 0.07% 0.04% 0.02% 0 IP ARP重试次数年龄
```

```
43 1703 16969 100 0.07% 0.00% 0.00% 0个IPC保活M
```

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发送流量对RP CPU的“IOSXE-RP平底船Se”，因此我们能调查进一步被踢对RP的此流量。

```
Router#show平台软件基础设施平底船
```

```
LSMPI接口内部stats：
```

```
enabled=0， disabled=0， throttled=0， unthrottled=0， 状态准备好
```

```
输入缓冲= 90100722
```

```
输出缓冲= 100439
```

```
rxdone count= 90100722
```

```
txdone count= 100436
```

Rx没有particletype count= 0
Tx没有particletype count= 0
从Shadow count= 0的Txbuf
数据包没有开始= 0
数据包没有末端= 0
平底船丢弃stats :
Bad版本0
Bad类型0
有功能报头0
有平台报头0
未命中0的功能报头
普通的报头不匹配0
Bad总长度0
Bad数据包长度0
Bad网络抵消了0
不是平底船报头0
未知链路类型0
没有swidb 1
坏ESS功能报头0
没有ESS功能0
没有SSLVPN功能0
我们的平底船类型未知0
平底船原因超出范围0

IOSXE-RP平底船数据包原因 :

62210226Layer2控制和传统数据包

147 ARP请求或响应数据包

27801234为我们数据包

84426 RP<->QFP keepalive数据包

6收集邻接数据包

1647为我们控制数据包

FOR_US控制IPv4 protocol stats :

1647个OSPF信息包

数据包histogram(500字节/bin), 在92的avg大小, 56 :

PAK大小计数外计数

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”, 此计数器通过多次查看命令确认增加在几分钟。这确认很多被踢的流量使用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, 当开始为根本原因时排除故障。例如, 数据包捕获或相似的交通监控可能实现发现被踢对将允许根本原因识别和解决防止在将来发生一个相似的问题的RP的确切的流量。

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地址通过配置Ip route发送ARP请求对广播接口。这可能也归结于从ARP表被冲洗的条目, 并且必须重学根据老化的MAC地址项或者接口出现/下来。

- **SSH** : 这可能导致高CPU由于一大show命令(show tech-support)或有很多调试命令启用强制很多CLI在SSH会话发送。
- **SNMP** : 这能归结于花些许多时间处理请求，并且导致高CPU的SNMP代理程序。经常两个可能原因是轮询的MIB，或者请路由并且/或者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连接飘荡。这可能指示路由去下来在网络内的或者沿着走强制的整个设备网络聚合和重新计算最佳路由，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识别高CPU的原因隔离，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

下列是其他另外的有用的命令列表使用，当排除故障高CPU和排序时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>** 在被踢的所有流量提供信息给RP
- **<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>** 当他们使用CPU，提供进程的一个实际更新的源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运行的所有进程摘要，以及平均负载
- **<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>** 当他们使用CPU，提供进程的一个实际更新的源在全局配置模式要求命令“terminal terminal-type”首先被输入正确地作用