

ASR 1000 シリーズ ルータ上の高い CPU 使用率のトラブルシューティング

内容

[概要](#)

[前提条件](#)

[要件](#)

[説明](#)

[トラブルシューティングの手順](#)

[手順 1：CPU の使用率が高いモジュールの特定](#)

[手順 2：モジュールの分析](#)

[手順 3：IOS プロセス](#)

[手順 4：Linux プロセス](#)

[手順 5：FECF プロセス](#)

[手順6:QFP使用率](#)

[手順 7：根本原因の特定と修正方法の識別](#)

[トラブルシューティングの例](#)

[追加コマンド](#)

[ルート プロセッサ](#)

[エンベデッド サービス プロセッサ](#)

概要

このドキュメントでは、ASR1000 シリーズ ルータでの CPU の高使用率の問題をトラブルシューティングする方法について説明します。

前提条件

要件

このドキュメントの解釈と利用には、[ASR1000アーキテクチャ](#)について理解しておくことをお勧めします。

説明

CiscoルータのCPU高使用率は、ルータのCPU使用率が通常の使用率を超えている状態と定義できます。一部のシナリオで CPU の使用率の増加があらかじめ予想されていても、別のシナリオでは同様の増加が問題と見なされる場合があります。ネットワークの変更や設定変更によるルータのCPU使用率の一時的な上昇は、無視して予期される動作になります。

ただし、ネットワークや構成を変更せずに長時間CPU使用率が高いルータは異常であるため、分析する必要があります。そのため、CPUが過剰な場合、他のすべてのプロセスをアクティブに処理できないため、コマンドラインの遅れ、コントロールプレーン遅延、パケット廃棄、サービス

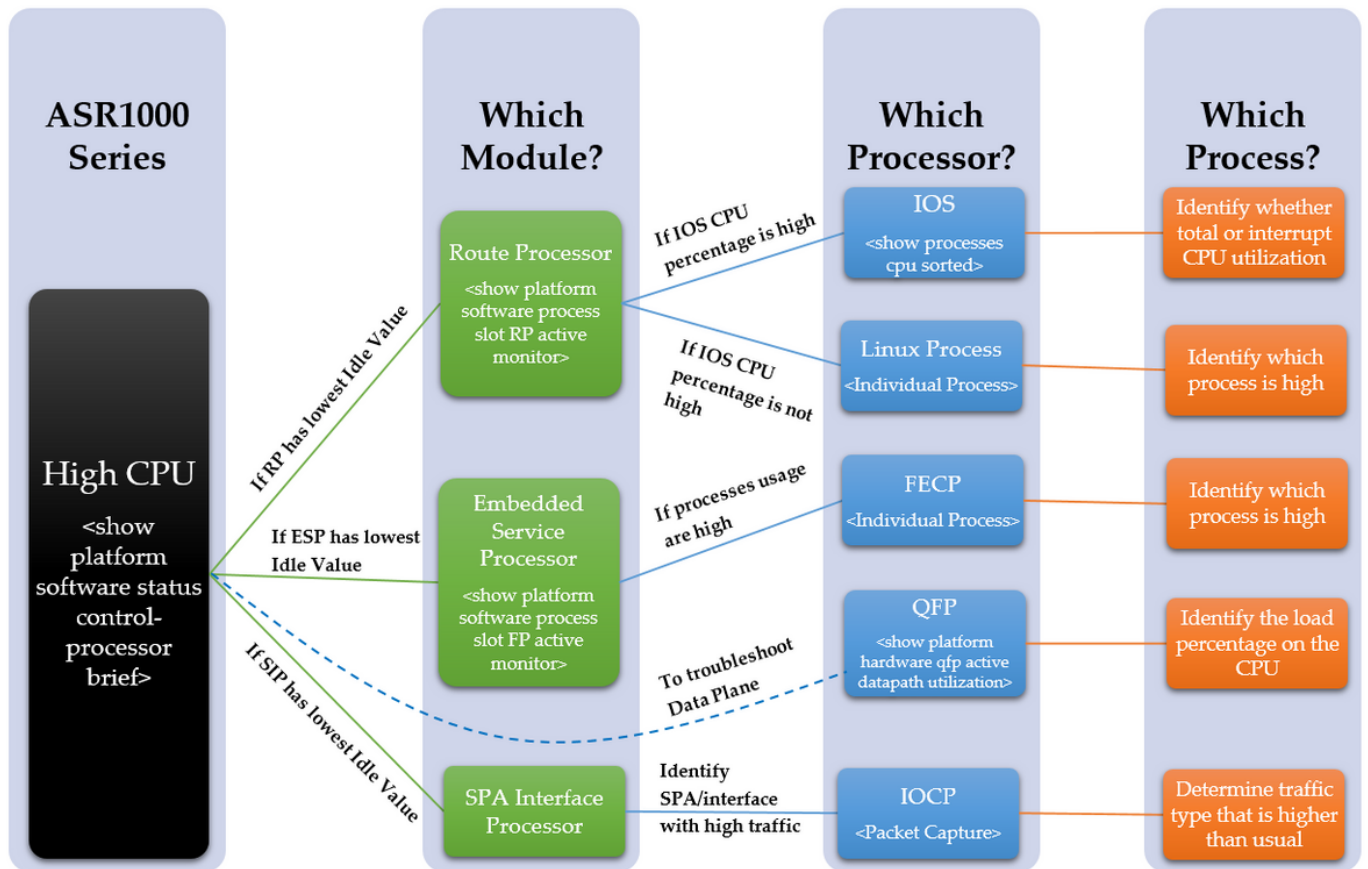
障害が発生します。

CPU の高使用率の理由として以下が考えられます。

1. コントロールプレーンCPUがパントトラフィックを受信し過ぎる
2. 予期せず動作し、CPU使用率が高くなるプロセス
3. データプレーンプロセッサが過剰に使用されている/オーバーサブスクライブされている
4. プロセッサの割り込みが多発する

ルータのCPU使用率はルータの負荷に比例するため、CPUの使用率が高くなることは必ずしもASR1000シリーズルータの問題とは限りません。たとえば、ネットワークが変更されると、ネットワークが再収束するため、コントロールプレーントラフィックが大量に発生します。したがって、CPUの過剰使用の根本原因を特定し、それが予想される動作か問題かを判断する必要があります。

次の図は、CPU高使用率の問題をトラブルシューティングする方法を段階的に説明したものです。



トラブルシューティングの手順

手順 1 : CPU の使用率が高いモジュールの特定

ASR1000は、モジュールごとに複数の異なるCPUを備えています。そのため、どのモジュールで、通常より使用率が高くなっているかを確認する必要があります。これは、アイドル値によって確認できます。アイドル値が小さいほど、そのモジュールのCPU使用率が高くなります。これらの異なるCPUはすべて、モジュールのコントロールプレーンを反映しています。

デバイス内のどのモジュールでCPUの使用率が高くなっているかを確認します。次のコマンドを

使用して、それが RP、ESP、SIP のいずれであるかを判断します。

```
show platform software status control-processor brief
```

強調表示された列を表示するには、次の出力を参照してください

RP のアイドル値が低くなっている場合、手順 2 のポイント 1 に進みます。

ESP のアイドル値が低くなっている場合、手順 3 のポイント 2 に進みます。

SIP のアイドル値が低くなっている場合、手順 4 のポイント 3 に進みます。

```
Router#show platform software status control-processor brief
```

```
Load Average
```

```
Slot      Status      1-Min  5-Min  15-Min
```

```
RP00.00 0.02 0.00
```

```
ESP0 Healthy 0.01 0.02 0.00
```

```
SIP00.00 0.01 0.00
```

```
Memory (kB)
```

```
Slot      Status      Total          Used (Pct)          Free
(Pct)      Committed (Pct)
```

```
RP02009376 1879196 (94%) 130180 (6%) 1432748 (71%)
```

```
ESP02009400 692100 (34%) 1317300 (66%) 472536 (24%)
```

```
SIP0471804 284424 (60%) 187380 (40%) 193148 (41%)
```

```
CPU Utilization
```

```
Slot      CPU    User    System  Nice  Idle      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
```

アイドル値がすべて比較的高ければ、コントロールプレーンの問題ではないと思われます。データプレーンをトラブルシューティングするには、ESP の QFP を監視する必要があります。QFP の使用率が高すぎてコントロールプレーンプロセッサの CPU 使用率が高くなることのないため、「CPU 使用率が高い」という症状が引き続き発生する可能性があります。手順 6 に進んでください。

手順 2 : モジュールの分析

• ルートプロセッサ

次のコマンドを使用して、RP 内で CPU 使用率が高いことが確認されたプロセッサを確認します。Linux プロセスでしょうか。それとも IOS でしょうか。

```
show platform software process slot RP active monitor
```

IOS CPU のパーセンテージが高くなっている場合 (linux_iosd-imag)、問題は RP IOS にあります。手順 3 に進んでください。

他のプロセスの CPU 使用率が高い場合は、Linux プロセスである可能性があります。手順 4 に進んでください。

・エンベデッド サービス プロセッサ

コントロールプレーンプロセッサのCPU使用率が高いことが確認された場合は、ESP内で確認します。それは FECP でしょうか。

```
show platform software process slot FP active monitor
```

プロセスの使用率が高くなっていけば、FECP に問題があります。ステップ 5 に進んでください。

FECPではない場合、ESP内の関連する問題を処理するコントロールプレーンではありません。ネットワーク遅延やキューのドロップなどの症状が引き続き発生する場合は、データプレーンの使用率を見直す必要があります。手順 6 に進んでください。

・SPA インターフェイス プロセッサ

SIPのCPU使用率が高いことが確認された場合、IOCPのCPU使用率が高いことが確認されます。IOCP内のどのプロセスでCPU使用率が高いことが確認されているかを確認します。

パケットキャプチャを実行し、通常よりも高いトラフィックと、このタイプのトラフィックに関連付けられているプロセスを特定します。ステップ7に進みます

手順 3 : IOS プロセス

次の出力を参照してください。最初のパーセンテージはCPUの総使用率で、2番目のパーセンテージは割り込みCPUの使用率で、これはパントされたパケットの処理に使用されるCPUの量です。

割り込みパーセンテージが高い場合は、大量のトラフィックがRPにパントされることを示します (show platform software infrastructure puntコマンドで確認できます)

割り込みパーセンテージが低く、合計CPU使用率が高い場合は、CPUを長期間にわたって使用するためのプロセスが観察されます。

次のコマンドを使用して、IOS内のどのプロセスが高いCPU使用率を示しているかを確認します。

```
show processes cpu sorted
```

どのパーセンテージ (全体の CPU または割り込みの CPU) が高くなっているか特定し、必要に応じて個別のプロセスを識別します。手順 7 に進んでください。

```
Router#show processes cpu sorted
```

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

```
PID(ms) 5515TTY
```

```
PID(ms) 5515TTY
```

```
188          8143          434758          18          0.15%    0.18%  
0.19%    0          Ethernet Msec Ti
```

```

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% 0 MMA DB TIMER

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

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

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

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

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

10 6 75 80 0.00% 0.00% 0.00% 0.00% 0

9          5671          538          10540          0.00%          0.14%
0.12%  0          Check heaps

```

手順 4 : Linux プロセス

IOSでCPUが過剰に使用されていることが確認された場合は、個々のLinuxプロセスのCPU使用率を確認する必要があります。これらのプロセスは、**show platform software process slot RP active monitor**からリストされたその他のプロセスです。CPUの使用率が高くなるプロセスを特定し、ステップ7に進みます。

手順 5 : FECP プロセス

プロセスの使用率が高い場合は、CPUの使用率が高くなる原因がFECP内のプロセスである可能性が高くなります。手順7に進みます

手順6:QFP使用率

Quantum Flow Processor はフォワーディング ASIC です。フォワーディングエンジンの負荷を判別するために、QFPをモニタできます。以下のコマンドによって、pps および bps 単位で入出力パケット (優先および非優先) がリストされます。最終行には、パケット転送による CPU 負荷の合計量がパーセンテージで示されます。

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

入力または出力が高いかどうかを確認し、プロセスの負荷を確認してから、ステップ7に進みます

```
Router#show platform hardware qfp active datapath utilization
```

```
CPP 0:Subdev 0 5 secs 1 min 5 min 60 min
```

```
Input: (pps) 0 0 0 0
```

```

                (bps) 208 176 176 176

(pps) 0 2 2 2

                (bps) 64 784 784 784

(pps) 0 2 2 2

                (bps) 272 960 960 960

(pps) 0 0 0 0

                (bps) 192 160 160 160

(pps) 0 1 1 1

                (bps) 0 6488 6496 6488

(pps) 0 1 1 1

                (bps) 192 6648 6656 6648

Processing: (pct) 0 0 0 0

```

手順 7：根本原因の特定と修正方法の識別

CPUの使用率が高すぎることを確認されたプロセスを使用すると、CPUの使用率が高くなった理由をより明確に把握できません。続行するには、特定されたプロセスによって実行される機能を調査します。これは、問題への対処方法に関するアクションプランを決定するのに役立ちます。例：プロセスが特定のプロトコルを担当している場合は、このプロトコルに関連する設定を確認できます。

それでもCPU関連の問題が発生する場合は、TACに連絡して、エンジニアがトラブルシューティングを進められるようにすることを推奨します。上記のトラブルシューティング手順は、エンジニアが問題をより効率的に切り分けるのに役立ちます。

トラブルシューティングの例

この例では、トラブルシューティングのプロセスを実行し、ルータの高CPU使用率の根本原因を最も適切に特定します。まず、CPUの使用率が高くなるモジュールを特定するために、次の出力を確認します。

```

Router#show platform software status control-processor brief
Load Average
Slot      Status      1-Min  5-Min  15-Min
RP00.66  0.15  0.05
ESP00.00  0.00  0.00
SIP00.00  0.00  0.00

Memory (kB)
Slot      Status      Total          Used (Pct)      Free
(Pct)    Committed (Pct)

```

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

CPU Utilization

```
Slot      CPU   User   System  Nice   Idle      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
```

RP0内のアイドル量が非常に少ないため、ルートプロセッサ内のCPUの高使用率の問題を示唆しています。したがって、さらにトラブルシューティングを行うために、RP内のどのプロセッサで高いCPU使用率が確認されているかを特定します。

```
Router#show processes cpu sorted
CPU utilization for five seconds:84%/36%;one minute:34%;five
minutes:9%
PID(ms)5515TTY
107      303230      50749      5975      46.69%      18.12%      4.45%
0      IOSXE-RP Punt Se
  63      105617      540091      195      0.23%      0.10%
0.08%  0      IOSD ipc task
 159 74792 2645991 28 0.15% 0.06% 0.06% 0 VRRS
 116      53685      169683      316      0.15%      0.05%
0.01%  0      Per-Second Jobs
  9      305547      26511      11525      0.15%      0.28%
0.16%  0      Check heaps
 188      362507      20979154      17      0.15%      0.15%
0.19%  0      Ethernet Msec Ti
  3      147      186      790      0.07%
0.08%  0.02%  0      Exec
  2      32126      33935      946      0.07%      0.03%
0.00%  0      Load Meter
 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 Retry Age
 43 1703 16969 100 0.07% 0.00% 0.00% 0 IPCM
```

この出力から、全体のCPUパーセンテージと割り込みのパーセンテージが予想されている量よりも多いことがわかります。CPUを使用する上位のプロセスは、RP CPUのトラフィックを処理するプロセスである「IOSXE-RP Punt Se」です。そのため、RPにパントされたトラフィックをさらに詳しく調べることができます。

```
Router#show platform software infrastructure punt
LSMPI interface internal stats:
enabled=0, disabled=0, throttled=0, unthrottled=0, state is ready
Input Buffers = 90100722
Output Buffers = 100439
rxdone count = 90100722
txdone count = 100436
Rx no particletype count = 0
```

```

Tx no particletype count = 0
Txbuf from shadow count = 0
No start of packet = 0
No end of packet = 0
Punt drop stats:
Bad version 0
Bad type 0
Had feature header 0
Had platform header 0
Feature header missing 0
Common header mismatch 0
Bad total length 0
Bad packet length 0
Bad network offset 0
Not punt header 0
Unknown link type 0
No swidb 1
Bad ESS feature header 0
No ESS feature 0
No SSLVPN feature 0
Punt For Us type unknown 0
Punt cause out of range 0
IOSXE-RP Punt packet causes:
  62 2102262
    147          ARP request or response packets
  27801234      For-us data packets
    84426      RP<->QFP keepalive packets
     6         Glean adjacency packets
    1647      For-us control packets

FOR_US Control IPv4 protcol stats:
  1647 OSPF packets
Packet histogram(500 bytes/bin), avg size in 92, out 56:
Pak-Size In-Count Out-Count
  0+:    90097805 98790
  500+:          0 7

```

この出力から、ルータに向かうトラフィックを示す「For-us data packets」に大量のパケットが含まれていることがわかります。このカウンタは、コマンドの観察から数分間にわたって増加していることが確認されています。これにより、多くの場合コントロールプレーントラフィックである大量のパントされたトラフィックによってCPUが過剰に使用されていることが確認されます。コントロールプレーントラフィックには、ARP、SSH、SNMP、ルートアップデート(BGP、EIGRP、OSPF)などがあります。この情報から、CPUの高使用率の潜在的な原因を特定し、根本原因のトラブルシューティングに役立てることが出来ます。たとえば、パケットキャプチャや異なるトラフィックのモニタを実装して、RPにパントされたトラフィックを正確に確認できます。これにより、根本原因を特定し、将来の同様の問題を回避できます。

パケットキャプチャを完了すると、潜在的なパントトラフィックの例は、次のようになります。

- **ARP** : これは、複数のIPアドレスがブロードキャストインターフェイスへのIPルートの設定を通じてARP要求を送信する場合に発生する過剰な数のARP要求が原因である可能性があります。これは、ARPテーブルからフラッシュされたエントリが原因である可能性があり、エンジニアリングアウトするMACアドレスエントリに基づいて再学習する必要があります。そうしな

いと、インターフェイスがアップ/ダウンします。

- **SSH** : これは、大きなshowコマンド(show tech-support)や、多くのdebugコマンドが有効にされているためにCPUの高使用率を引き起こす可能性があり、これによりSSHセッション経由で大量のCLIが強制的に送信されます。
- **SNMP** : これは、SNMPエージェントが要求の処理に長い時間がかかるため、CPUの使用率が高いことが原因である可能性があります。多くの場合、2つの考えられる原因は、ポーリングされるMIB、またはNMSによってポーリングされるルートテーブルとARPテーブルです。
- **ルートアップデート** : ルートアップデートの流入は、ネットワークの再コンバージェンスやリンクフラップが原因であることが多いです。これは、ネットワーク内でダウンするルート、またはネットワークを強制的に収束させ、最適なルートを再計算させるダウンするデバイス全体を示す可能性があります。これは、使用されているルーティングプロトコルによって異なります。

これは、個々のプロセスレベルに低下した場合に、CPUの高使用率の原因を特定することによって根本原因を特定する方法を強調します。ここから、発生している状況が設定の問題であるか、ソフトウェアの問題であるか、またはネットワーク設計に起因しているか、それとも意図された動作であるかを識別するために、個々のプロセスまたはプロトコルを切り離して分析することができます。

追加コマンド

次に、その他の便利なコマンドのリストを示します。これらのコマンドは、どのプロセッサに関連しているかによってソートされます。

ルート プロセッサ

- `<show process cpu history>` 過去 60 秒、分、72 時間の CPU の履歴をグラフで示します。
- `<show process process_ID>` 個々のプロセスに割り当てられているメモリおよび CPU の詳細を示します。
- `<show platform software infrastructure punt>` RPにパントされるすべてのトラフィックに関する情報を提供します。
- `<show platform software status control-processor brief>` メモリとモジュールの統計情報とともに CPU の負荷と「健全性」を示します。
- `<show platform software process slot r0|r1 monitor>` 選択したモジュールのプロセスおよびそれらの CPU とメモリの割り当てに関する詳細を示します。
- `<monitor platform software process r0|r1>` CPUを使用してプロセスを更新するライブフィードを提供正常に機能するように、最初にグローバル コンフィギュレーション モードでコマンド「terminal terminal-type」を入力する必要があります。

エンベデッド サービス プロセッサ

- `<show platform software process list fp active summary>` CPUで実行されているすべてのプロセスの概要と平均負荷の詳細を表示します
- `<show platform software process slot f0|f1 monitor>` 選択したモジュールのプロセスおよびそれらの CPU とメモリの割り当てに関する詳細を示します。
- `<monitor platform software process f0|f1>` CPUを使用するプロセスを更新するライブフィー

ドを提供します正常に機能するように、最初にグローバル コンフィギュレーション モードで
コマンド「terminal terminal-type」を入力する必要があります。