

ASR1000 シリーズ ルータの高CPU を解決して下さい

目次

[概要](#)

[前提条件](#)

[要件](#)

[説明](#)

[トラブルシューティングの手順](#)

[ステップ 1-高CPU のモジュールを識別して下さい](#)

[ステップ 2-モジュールを分析して下さい](#)

[ステップ 3- IOSプロセス](#)

[ステップ 4- Linux プロセス](#)

[ステップ 5- FECF プロセス](#)

[ステップ 6- QFP 利用](#)

[ステップ 7-根本的な原因を判別し、修正を識別して下さい](#)

[例を解決して下さい](#)

[追加コマンド](#)

[ルート プロセッサ](#)

[組み込みサービス処理機構](#)

概要

この資料に ASR1000 シリーズ ルータの高CPU 問題を解決する方法を記述されています。

前提条件

要件

Cisco は [ASR1000 アーキテクチャ](#) のよい理解があることを推奨します。

説明

Ciscoルータの高CPU はルータの CPU稼働率が正常な使用方法の上にある条件と定義されるかもしれませんが。あるシナリオでは増大したCPU 使用方法はもう1つのシナリオでは問題を示唆する可能性がある間、期待されます。ネットワーク変更かコンフィギュレーション変更によるルータの一時 CPU使用率が高い状態は無視することができ、期待されます。

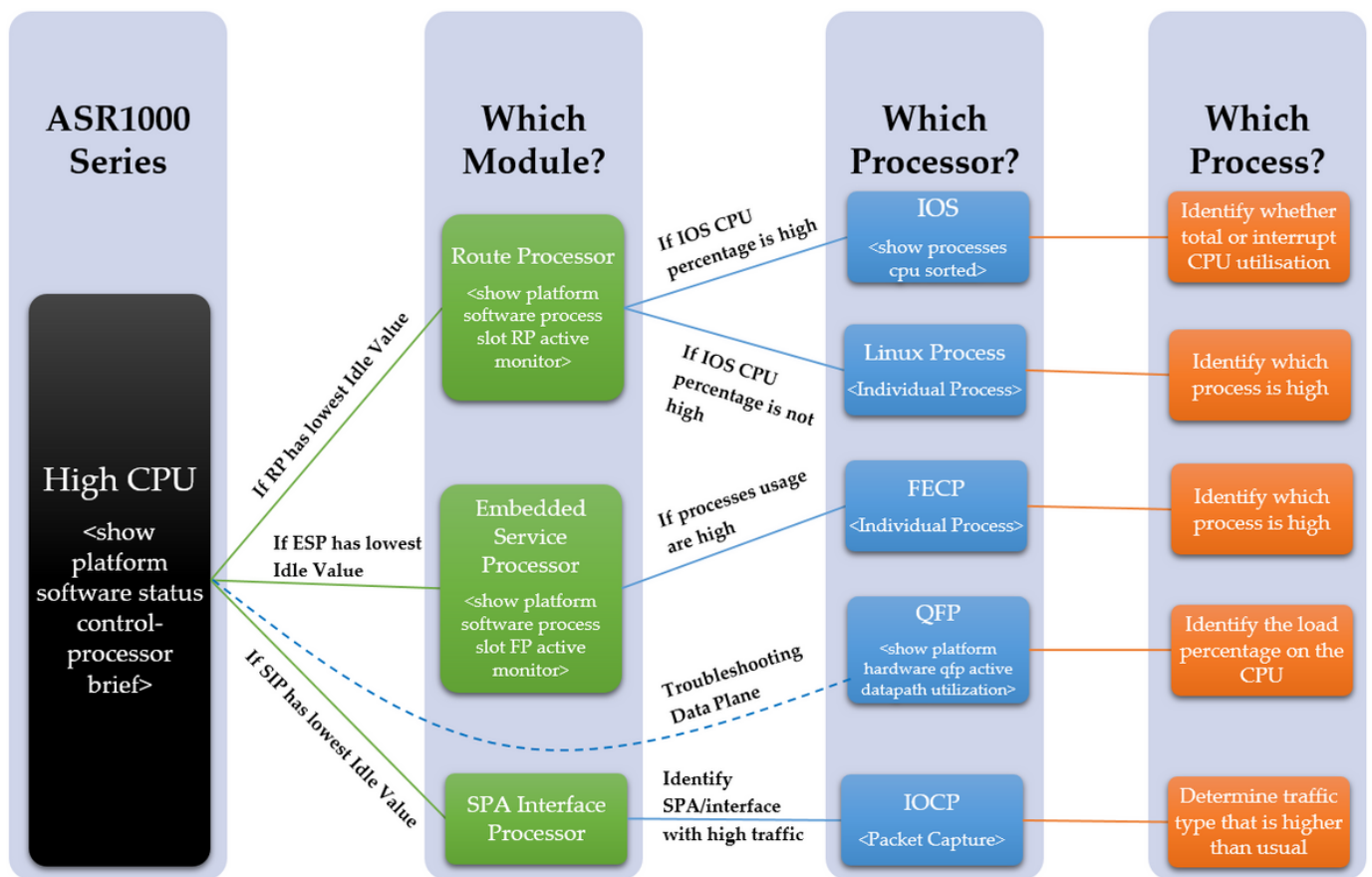
ただしネットワークまたは設定の変更なしで長期のための CPU使用率が高い状態を経験するルータは珍しく、分析される必要があります。この過剰使用はアクティブに遅いコマンド・ライン、コントロールプレーンレイテンシー、パケット破棄および壊れるサービスに終って他のプロセスをすべて、保守できない CPU に導きます。

高CPUの原因は次のとおりです:

1. コントロールプレーン CPU は余りにも多くのパケットトラフィックを受信します
2. CPU を必要以上に使用する不品行な振舞いをうプロセス
3. データプレーンプロセッサは利用された /oversubscribed にあります
4. 余りにも多くのプロセッサの割り込み

高CPU はルータCPU利用率がルータのロードに正比例しているため ASR1000 シリーズ ルータ 問題常にではないです。たとえばネットワーク変更があれば、ネットワークが再収束するようにこれにより多量のコントロールプレーントラフィックを引き起こします。従ってそれが予期された動作または問題だったかどうかを確認する、利用上の CPU の根本的な原因を判別する必要があります。

infographic 下記にありま方法のステップバイステッププロセスを高CPU問題を解決する詳述します:



トラブルシューティングの手順

ステップ 1 -高CPU のモジュールを識別して下さい

ASR1000 に異なるモジュールを渡る複数の CPU があります。従って、どのモジュールが非常により正常な使用方法を表示するか参照する必要があります。これはアイドル状態の値によって、ように下部のアイドル状態の値見られる場合があります、より高いそのモジュールの CPU 利用。これらの 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ます。

デバイス内のどのモジュールが高CPUを経験しているか判別して下さい。それは下記のコマンドで RP、ESP、または SIP あります

プラットフォームソフトウェア ステータス *Control Processor* 要約を示して下さい

強調表示されたカラムを表示するために図 1 参照して下さい

RP に低いアイドル状態の値がある場合、ステップ 2 ポイント 1 に進んで下さい

ESP に低いアイドル状態の値がある場合、ステップ 3 ポイント 2 に進んで下さい

SIP に低いアイドル状態の値がある場合、ステップ 4 ポイント 3 に進んで下さい

Router#show プラットフォームソフトウェア ステータス *Control Processor* 要約

ロード平均

スロット ステータス 1 最小値 5 最小値 15 最小値

RP0 健全な 0.00 0.02 0.00

ESP0 健全な 0.01 0.02 0.00

SIP0 健全な 0.00 0.01 0.00

メモリ (kb)

使用されるスロット ステータス合計 (Pct) 自由に (Pct) 託される (Pct)

RP0 健全な 2009376 1879196 (94%) 130180 (6%) 1432748 (71%)

ESP0 健全な 2009400 692100 (34%) 1317300 (66%) 472536 (24%)

SIP0 健全な 471804 284424 (60%) 187380 (40%) 193148 (41%)

CPU 使用率

スロット CPU ユーザシステム ニース **アイドル状態** 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

アイドル状態の値がすべての比較的高い場合、それはコントロールプレーン問題ではないかもしれませんが。データを解決するために ESP の QFP を観察される必要があります平にして下さい。「高CPU」の現象はまだ利用されるにある高CPUを報告するコントロールプレーンプロセッサという結果に終わらない QFP が観察された原因である場合もあります。ステップ 6 に進みます。

ステップ 2 –モジュールを分析して下さい

• ルート プロセッサ

プロセッサが下記のコマンドを使用して高CPUを経験している RP の内で確認して下さい。それは Linux プロセスまたは IOS ですか。

プラットフォームソフトウェア プロセス スロット RP アクティブ モニタを示して下さい

IOS CPU パーセントが高ければ (linux_iosd-imag)、です RP IOS あります。STEP 3 に進んで下さい

他のプロセス CPU パーセントが高い場合、Linux プロセスであることは可能性が高いです。STEP 4 に進んで下さい

• 組み込みサービス処理機構

コントロールプレーンプロセッサが高CPUを経験しているかどうか ESP の内で確認して下さい。それは FECP ですか。

プラットフォームソフトウェア プロセス スロット FP アクティブ モニタを示して下さい

プロセスが高ければそれは FECP、STEP 5 に進みます

FECP でなければそれは ESP 内の問題を処理するコントロールプレーンではないです。ネットワーク待ち時間のような現象がキュードロップがそれでも観察される場合、データ平面は利用のために検討される必要がある場合もあります。STEP 6 に進んで下さい

• SPA インターフェイス プロセッサ

SIP が高CPU を経験すれば IOCP は高CPU を経験しています。IOCP 内のプロセスかプロセスが CPU を最も利用しているかどれ判別して下さい

パケットキャプチャを行い、どのトラフィックが通常より高く、どのプロセスがこのトラフィックの種類と関連付けられるか特定して下さい

識別して下さいプロセスかプロセスが CPU を最も利用している。STEP 7 に進んで下さい

ステップ 3 – IOS プロセス

図 2 参照して下さい、最初のパーセントは合計 CPU 利用であり、第 2 パーセントはパントされたパケットを処理するために使用される CPU の量の割り込み CPU 利用です、

多量のトラフィックは RP にパントされていることを示すこと割り込みパーセントが高ければ、(これは示しますプラットフォームソフトウェア インフラストラクチャ パントをと確認することができます)

割り込みパーセントが低い、合計 CPU が高い場合、期待されるに CPU を長期間にわたって利用するプロセスまたはプロセスがあります。

プロセスかプロセスが下記のコマンドを使用して高CPU 利用を経験している IOS の内で確認して下さい。

ソートされる *show processes cpu*

どのパーセントが高い (総 CPU はまたは CPU を割り込みます) 識別し、次に必要であれば個々のプロセス/プロセスをか特定して下さい。STEP 7 に進んで下さい

Router#show はソートされる CPU を処理します

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

PID Runtime (ms) は uSecs 5Sec 1Min 5Min TTY Process を呼び出しました

PID Runtime (ms) は uSecs 5Sec 1Min 5Min TTY Process を呼び出しました

188 8143 434758 18 0.15% 0.18% 0.19% 0 イーサネット ミリ秒 Ti

515 380 7050 53 0.07% 0.00% 0.00% 0 SBC 主要なプロセス

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

380 1783 55002 32 0.07% 0.06% 0.06% 0 MUTTAHIDA MAJLIS-E-AMAL DB タイマー

63 3132 11143 281 0.07% 0.07% 0.07% 0 IOSD IPC タスク

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

6 19 12 1583 0.00% 0.00% 0.00% 0 RF スレーブ 主要な Th

8 0 1 0 0.00% 0.00% 0.00% 0 RO 呼出 タイマー

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

10 6 75 80 0.00% 0.00% 0.00% 人の 0 人のプール マネージャ

9 5671 538 10540 0.00% 0.14% 0.12% 0 チェック ヒープ

ステップ 4 – Linux プロセス

IOS が CPU を必要以上に使用しない場合、これらのプロセスです提示プラットフォームソフトウェア プロセス スロット RP アクティブ モニタからリストされている他のプロセス個々の Linux プロセスのための CPU稼働率を検知する必要があります。識別して下さいプロセスかプロセスが高CPUを経験したりそして STEP 7 に進む

ステップ 5 – FECP プロセス

プロセスかプロセスが高かったらそれからそれ本当らしかったらそれら高CPU利用を引き起こしている FECP 内のプロセス、STEP 7 に進んで下さい

ステップ 6 – QFP 利用

Quantum フロー プロセッサは転送 ASIC です。QFP を監視することはフォワーディングエンジンのロードを明らかにします。下記のコマンド リスト パケット毎秒の入出力パケット (優先順位および非優先)、およびビット/秒。最終的なライン表示パーセントのパケット転送による CPU負荷の総量。

プラットフォーム ハードウェア qfp にアクティブな datapath 利用を示して下さい

入力か出力が高かったら識別し、処理負荷を表示し、次に STEP 7 に進んで下さい

Router#show プラットフォーム ハードウェア qfp アクティブな datapath 利用

CPP 0: Subdev 0 5 秒 1 最小値 5 最小値 60 最小値

Input: 優先順位 (pps) 0 0 0 0

(ビット/秒) 208 176 176 176

非優先 (pps) 0 2 2 2

(ビット/秒) 64 784 784 784

合計 (pps) 0 2 2 2

(ビット/秒) 272 960 960 960

出力 : 優先順位 (pps) 0 0 0 0

(ビット/秒) 192 160 160 160

非優先 (pps) 0 1 1 1

(ビット/秒) 0 6488 6496 6488

合計 (pps) 0 1 1 1

(ビット/秒) 192 6648 6656 6648

処理: ロード (pct) 0 0 0 0

ステップ 7 –根本的な原因を判別し、修正を識別して下さい

識別される CPU を必要以上に使用しているプロセスかプロセスを使ってクリア ピクチャがある高CPU がなぜのプロセッサの発生したりまたは過剰利用か。特定されたプロセスによって実行された 機能を研究して下さい。これは私達に方法のアクションプランを問題にアプローチする与えます。たとえば-プロセスが特別なプロトコルに責任があればその設定を検知したいと思う場合もあります。

まだ問題に直面する場合、エンジニアが更に解決するのを助けることを可能にするように TAC に接触することを推奨します。上記のトラブルシューティング の手順はエンジニアが問題をより簡単に特定するのを助けます。

例を解決して下さい

この例でトラブルシューティング プロセスによって動作し、推奨に試みるためにルータ 高CPU のための可能性のある 根本的な原因を特定して下さい。どのモジュールが高CPU を経験しているか識別から開始して、次の出力があります:

Router#show プラットフォームソフトウェア ステータス Control Processor 要約

ロード平均

スロット ステータス 1 最小値 5 最小値 15 最小値

RP0 健全な 0.66 0.15 0.05

ESP0 健全な 0.00 0.00 0.00

SIP0 健全な 0.00 0.00 0.00

メモリ (kb)

使用されるスロット ステータス合計 (Pct) 自由に (Pct) 託される (Pct)

RP0 健全な 2009376 1879196 (94%) 130180 (6%) 1432756 (71%)

ESP0 健全な 2009400 692472 (34%) 1316928 (66%) 472668 (24%)

SIP0 健全な 471804 284556 (60%) 187248 (40%) 193148 (41%)

CPU 使用率

スロット CPU ユーザシステム ニース アイドル状態 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
```

従って RP0 内のアイドル状態の量は非常に低いので、RP 内のどのプロセッサが高CPU を経験しているか私達を解決し続けるようにルートプロセッサ内の高CPU 問題を、識別します提案します。

Router#show はソートされる CPU を処理します

```
CPU utilization for five seconds: 84%/36%; one minute: 34%; five minutes: 9%
PID Runtime ( ms ) は uSecs 5Sec 1Min 5Min TTY Process を呼び出しました
107 303230 50749 5975 46.69% 18.12% 4.45% 0 IOSXE-RP パント Se
63 105617 540091 195 0.23% 0.10% 0.08% 0 IOSD IPC タスク
159 74792 2645991 28 0.15% 0.06% 0.06% 0 VRRS メイン スレッド
116 53685 169683 316 0.15% 0.05% 0.01% 0 毎秒ジョブ
9 305547 26511 11525 0.15% 0.28% 0.16% 0 チェック ヒープ
188 362507 20979154 17 0.15% 0.15% 0.19% 0 イーサネット ミリ秒 Ti
3 147 186 790 0.07% 0.08% 0.02% 0 Exec
2 32126 33935 946 0.07% 0.03% 0.00% 0 ロードメータ
446 416 33932 12 0.07% 0.00% 0.00% 0 VDC プロセス
164 59945 5261819 11 0.07% 0.04% 0.02% 0 IP ARP 再試行経過時間
43 1703 16969 100 0.07% 0.00% 0.00% 0 IPC キープ アライブ M
```

この出力から、それは合計 CPU パーセントおよび割り込みパーセントが予想を越えたであること観察することができます。CPU を利用する上プロセスはプロセスの「IOSXE-RP パント Se」です従って RP CPU ヘトラフィックを送信 することを処理する、RP にパントされるこのトラフィックに更に検知できます。

Router#show プラットフォームソフトウェア インフラストラクチャ パント

LSMPI インターフェイス 内部統計:

enabled=0 は、disabled=0、throttled=0、unthrottled=0、状態準備ができています

インプットバッファ = 90100722

アウトプットバッファ = 100439

rxdone 数 = 90100722

txdone 数 = 100436

Rx particletype 数無し = 0

Tx particletype 数無し = 0

シャドウ数からの Txbuf = 0

パケットの開始する無し = 0

パケットの終わり無し = 0

パント ドロップする統計:

悪いバージョン 0

悪い型 0

機能 ヘッダ 0 を持っていました

プラットフォーム ヘッダ 0 を持っていました

0 が抜けている機能 ヘッダ

よくあるヘッダ ミスマッチ 0

悪い全長 0

悪いパケット 長 0

悪いネットワークは 0 を相殺しました

ないパント ヘッダ 0

未知リンクタイプ 0

swidb 無し 1

悪い ESS 機能 ヘッダ 0
ESS 機能無し 0
SSLVPN 機能無し 0
私達のためのパント型未知数 0
範囲 0 からのパント原因

IOSXE-RP パント パケット原因:
62210226Layer2制御およびレガシーパケット
147 ARP要求かレスポンスパケット
27801234 のため私達データパケット
84426 の RP<->QFP キープアライブ パケット
6 つのグリーンング隣接関係パケット
1647 のため私達制御パケット

FOR_US コントロール IPv4 protocol 統計:
1647 の OSPF パケット
パケット histogram(500 バイト/ビン)、92 の avg サイズ、56:
朴サイズの内部数数
0+: 90097805 98790
500+: 0 7

この出力から、数分ことをにわたるコマンド複数回の表示によってルータの方に送信されるトラフィックを示すに多量のパケットが「のため私達データパケット」このカウンター増分するために確認されたあることを見る場合があります。これは頻繁にコントロールプレーントラフィックである CPU が多量のパントされたトラフィックによって利用されていることを確認します。コントロールプレーントラフィックは ARP、SSH、SNMP、ルーティング更新 (BGP、EIGRP、OSPF) 先祖などを含むことができます この情報から、高CPU の潜在的な原因を特定でき、始まる時これは根本的な原因のために解決することを助けます。将来起こる同じような問題を防ぐために根本的な原因が特定され、解決するようにする RP にパントされる正確なトラフィックを見るためにたとえば、パケットキャプチャが同じような交通モニタリングは設定できません。

パケットキャプチャが完了すれば、可能性によってパントされるトラフィックの例は次のとおりです:

- **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に IP ルートの設定によって ARP要求を送信した場合発生する過剰 ARP要求が原因である可能性があります。これはまた ARPテーブルからフラッシュされるエントリが原因である可能性があり、基づいていました老化する MACアドレスエントリがインターフェイス/にアップ学び直されなければならなくて。
- **SSH:** これにより大きい表示コマンド (show tech-support に) による高CPU を引き起こす可能性がありますまたは多くのデバッグコマンドを持っていることは多くの CLI を SSH セッションに送信されるために強制することを有効にしました。
- **SNMP :** これは要求を処理するのに時間をかけるおよび従って高CPU を引き起こす SNMP エージェントが原因である可能性があります。頻繁に 2 つの推定 原因はポーリングされる MIB ですまたはおよび/または NMS によってポーリングされる ARPテーブル ルーティングして下さい。
- **ルーティング更新:** 多くの場合ルーティング更新の流入はネットワーク再収束が原因で、また

はフラッピングをリンクします。これはコンバージし、ルーティング プロトコルが使用中である最良の ルートを計算し直すために行くネットワークの中で、かネットワークの強制をたどって行く全体のデバイスを示す可能性がありますルーティング。

これは個々の プロセス レベルに来るとき根本的な原因が高CPU の原因の特定によってどのように特定することができるか強調表示します。ここから、個々の プロセスかプロトコルは分離でそれが設定に関する 問題、ソフトウェア上の問題、ネットワーク設計、または意図されていた推奨事項であるかどうか識別するために分析することができます。

追加コマンド

以下はプロセッサによってに関連している高CPU をトラブルシューティングするときソートされる利用する他の追加役に立つコマンドのリストであり、：

ルート プロセッサ

- `<show プロセス CPU history>` 最後の 60 秒、分および 72 時間にわたる CPU 履歴のグラフを提供します
- `<show プロセス process_ID>` 個々の プロセス メモリおよび CPU アロケーションの詳細な情報
- `<show プラットフォームソフトウェア インフラストラクチャ punt>` RP にパントされるすべてのトラフィックで情報を提供します
- `<show プラットフォームソフトウェア ステータス Control Processor brief>` CPU のロードおよび「健全性を詳述しましたり、またメモリおよびモジュール統計情報を詳述します
- `<show プラットフォームソフトウェア プロセス スロット r0|r1 monitor>` 選択したモジュールの異なるプロセスおよび CPU およびメモリ 割り当て詳述します
- `<monitor プラットフォームソフトウェア プロセス r0|r1>` それらが CPU を利用していると同時にプロセスのライブ アップデート供給を提供しますコマンド「ターミナル ターミナル型」がグローバル コンフィギュレーション モードで正しく機能するために最初に入力されるように要求します

組み込みサービス処理機構

- `<show プラットフォームソフトウェア プロセス リスト fp アクティブ summary>` CPU で動作するすべてのプロセスの要約、また平均負荷を詳述します
- `<show プラットフォームソフトウェア プロセス スロット f0|f1 monitor>` 選択したモジュールの異なるプロセスおよび CPU およびメモリ 割り当て詳述します
- `<monitor プラットフォームソフトウェア プロセス f0|f1>` それらが CPU を利用していると同時にプロセスのライブ アップデート供給を提供しますコマンド「ターミナル ターミナル型」がグローバル コンフィギュレーション モードで正しく機能するために最初に入力されるように要求します