

Cisco ルータの CPU 使用率が高い場合のトラブルシューティング

目次

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[CPU 使用率が高い場合の症状](#)

[最初のトラブルシューティング](#)

[原因の特定と問題の解決](#)

[CPU 高使用率の原因が割り込みにある場合](#)

[Netflow NDE 7600 シリーズ ルータを on Cisco 有効にする場合の高CPU](#)

[プロセスが原因でCPUが高負荷となる場合](#)

[PCI およびファースト メモリ プールに非常に高い使用率が示される場合](#)

[%SNMP-4-HIGHCPU: プロセスは\[DEC\] ms しきい値 \(\[DEC\] ms IOS クォンタム \) をのための\[文字\]の超過します\[文字\]---結果\[文字\]](#)

[ソフトウェア暗号化による高CPU](#)

[フラグメンテーションによる CPU使用率が高い状態](#)

[詳細情報を取得するためのコマンド](#)

[show processes cpu コマンド](#)

[show interfaces コマンド](#)

[show interfaces switching コマンド](#)

[show interfaces stat コマンド](#)

[show ip nat translations](#)

[show align コマンド](#)

[show version コマンド](#)

[show log コマンド](#)

[定期的にデータを収集するための UNIX シェル スクリプト](#)

[関連情報](#)

概要

このドキュメントでは、Cisco ルータで CPU 使用率が高いときの一般的な症状、原因、および解決策について説明しています。

前提条件

要件

Cisco ルータで CPU の使用率が高い場合のトラブルシューティングを行うには、Cisco IOS(R) ソフトウェアのスイッチング パスを理解しておく必要があります。Cisco IOS ソフトウェアのス

イッチングパスについての詳細は、『[パフォーマンスチューニングに関する基本事項](#)』を参照してください。

使用するコンポーネント

このドキュメントは、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。

CPU 使用率が高い場合の症状

次のリストは、CPU 使用率が高い場合の一般的な症状を示しています。これらの症状のいずれかが見られる場合は、このドキュメントのトラブルシューティング手順に従って問題を軽減させてください。

- `show processes cpu` コマンドの出力での高いパーセンテージ表示Ciscoデバイスからの `show processes cpu` コマンドの出力がある場合、潜在的な問題および修正を表示するのに [Cisco CLI アナライザ](#)を使用できます。 [Cisco CLI アナライザ](#)を使用するために、[登録 ユーザ](#)である必要があります。ログオンされ、JavaScript を有効にしてもらいます。
- パフォーマンスが悪い
- ルータ上のサービスが正常に応答しない。たとえば、次のような症状が起こる。Telnet の応答が遅い、またはルータに Telnet できないコンソールの応答が遅いping の応答が遅い、またはまったく応答しないルータが他のルータにルーティング アップデートを送信しない
- 高いバッファ障害

最初のトラブルシューティング

の [CPU使用率が高い状態の現象](#)からの現象注意すれば:

- セキュリティ問題が発生していないかどうかをチェックします。一般に、CPU の高使用率は、ネットワーク内で活動するワームやウイルスなどのセキュリティ問題によって発生します。たとえば、最近ネットワークに対する変更を行っていない場合は、これが原因になっている可能性が高くなります。通常は、アクセスリストに行を追加するといった設定の変更によって、この問題の影響を軽減できます。「[Cisco 製品セキュリティアドバイザリおよび注意](#)」ページには、一番可能性が高い原因の検出方法と、具体的な回避策が記載されています。詳細については、次を参照してください。[インターネットの脅威に関する 100 の質問と回答](#)
[シスコ製品のセキュリティについての勧告および注意Cisco 脅威管理](#)
- `undebg all` コマンドまたは `no debug all` コマンドを発行して、必ずルータのすべてのデバッグ コマンドを無効にしておきます。デバッグ コマンドの使用方法については、『[debug コマンドの使用法](#)』を参照してください。
- ルータで `show` コマンドを発行できるかどうかを確認します。発行できる場合は、これらの [show](#) コマンドを使用して、すぐに情報の収集を開始します。
- ルータへのアクセス可否を確認します。この問題に再現性があるかどうかを確認します。再現性がある場合は、ルータの電源をいったんオフにしてから再度オンにして、[scheduler interval 500](#) コマンドを設定してから問題を再現します。これにより、優先順位の低いプロセスが 500 ミリ秒ごとに実行されるようスケジュールされるため、CPU 使用率が 100 % であっても、いくつかのコマンドを実行できます。Cisco 7200 および Cisco 7500 シリーズルータでは、[scheduler allocate 3000 1000](#) コマンドを使用します。

- CPU 高負荷時の症状が、短く不規則な間隔で発生しているかを確認します。そうである場合は、[show processes cpu](#) コマンドの出力を一定の間隔で収集します。これにより、CPU 高負荷の原因が割り込みにあるのか、または特定のプロセスにあるのかがわかります。この [UNIX スクリプト](#) を使用し、最初の観察結果に基づいてこのスクリプトを変更して、問題の調査を進めるために必要なデータを収集します。

原因の特定と問題の解決

[show processes cpu](#) コマンドを使用して、CPU の使用率が高くなっている原因が割り込みにあるか、プロセスにあるかをチェックします。

CPU 高使用率の原因が割り込みにある場合

詳細は、『[割り込みによって CPU 使用率が高くなる場合のトラブルシューティング](#)』を参照してください。割り込みによって CPU 使用率が高くなる原因が CEF のパケット交換による可能性が高い場合、CPU の使用率によるルータ パフォーマンスへの影響はありません。

Netflow NDE 7600 シリーズ ルータを on Cisco 有効にする場合の高CPU

NetFlow がバージョン 7 のために設定される場合、フローは CPU 使用率が高い状態を引き起こす可能性があるルーティング プロセッサによって実行された。

Netflow バージョン 7 による CPU 使用率が高い状態のトラブルシューティングのために NetFlow エクスポートがバージョン 5 またはバージョン 9 のためのデフォルトである SP によって実行されたように、[MLS nde 送信側](#) バージョン 5 を設定して下さい。

プロセスが原因でCPUが高負荷となる場合

どのプロセスが CPU に負荷を与えているかをチェックします。プロセスについて通常とは異なるアクティビティがあれば、ログにエラー メッセージが記録されます。したがって、まず大量の CPU サイクルを消費しているプロセスに関連するエラーがないかどうかを [show logging](#) exec コマンドの出力でチェックします。

またデバッグは、プロセスによる CPU の高使用率をトラブルシューティングするために非常に役立ちます。ただし、デバッグを行うことで CPU 使用率がより高くなるため、デバッグを実行する際には細心の注意が必要です。安全で有用なデバッグを行うには、これらの前提条件を満たす必要があります。

- 適切な [logging destination \[severity-level\]](#) 設定コマンドを使用して、バッファ ログ以外ログの出力先をディセーブルにするか、またはそのログ重大度を 7 (デバッグ) から 6 (情報提供) またはそれ以下に設定します。イネーブルになっているログの出力先と対応レベルは、[show logging](#) exec コマンドのヘッダー行に示されています。
- ログ バッファのサイズは、十分な情報を取得できるように増加しておく必要があります。詳細は、[logging buffered](#) グローバル設定コマンドの説明を参照してください。
- より的確な情報をデバッグで得るために、日時とミリ秒のタイムスタンプをイネーブルにしておきます。詳細は、[service timestamps](#) グローバル設定コマンドの説明を参照してください。

『[プロセスによって CPU 使用率が高くなる場合のトラブルシューティング](#)』には、IP パケットのデバッグ セッション例が示されています。

特定のプロセスでの CPU 高使用率のトラブルシューティングについては、次の該当するリンクを参照してください。

- [ARP Input](#) : ドキュメント『プロセスによって CPU 使用率が高くなる場合のトラブルシューティング』の「ARP Input」セクション
- [BGP ルータ](#) : ドキュメント『トラブルシューティング : BGP スキャナまたは BGP ルータ プロセスが原因で発生する CPU の高使用』の「CPU 高使用の原因が BGP ルータ プロセスにある場合」セクション
- [BGP スキャナ](#) : ドキュメント『トラブルシューティング : BGP スキャナまたは BGP ルータ プロセスが原因で発生する CPU の高使用』の「CPU 高使用の原因が BGP スキャナ プロセスにある場合」セクション
- [EXEC — EXEC](#) および仮想実行プロセスの CPU 使用率が高い状態
- [HyBridge Input](#) : 『ATM インターフェイスが備わるルータでの、Hybridge 入力プロセスによる CPU 高使用率のトラブルシューティング』
- [IP Input](#) : 『IP インプット プロセスにより CPU 使用率が高い場合のトラブルシューティング』
- [IP の簡易ネットワーク管理プロトコル \(SNMP\)](#) : 『IP の簡易ネットワーク管理プロトコル (SNMP) によって生じる CPU の高使用率』
- [LC ADJ Updater](#) : 『Cisco 12000 シリーズ インターネット ルータの LC 隣接関係更新処理で CPU の使用率が高くなることの原因』
- [TCP Timer](#) : ドキュメント『プロセスによって CPU 使用率が高くなる場合のトラブルシューティング』の「TCP Timer」セクション
- [TTY Background](#) : ドキュメント『プロセスによって CPU 使用率が高くなる場合のトラブルシューティング』の「TTY バックグラウンド」セクション
- [Virtual EXEC](#) : 『EXEC および仮想 EXEC プロセスでの CPU 高使用率』
- [Vtemplate Backgr](#) : ドキュメント『プロセスによって CPU 使用率が高くなる場合のトラブルシューティング』の「仮想テンプレート バックグラウンド」セクション
- SSH プロセス— `show tech` がデバッグをキャプチャ することが有効になる場合高く行くかもしれません。
- [その他のプロセス](#) : ドキュメント『プロセスによって CPU 使用率が高くなる場合のトラブルシューティング』の「他のプロセス」セクション

PCI およびファースト メモリ プールに非常に高い使用率が示される場合

PCI およびファースト メモリ プールで空きメモリの低下が見られることは正常です。PCI メモリは、接続されている PCI バスの PRP メインボード上の GT64260 コントローラに対するメモリ アクセスに使用されます。このメモリは、システム コントローラと他の部分の間の内部通信に使用されるため、常に使用率が高いように見えます。

さらに多くのメモリが必要な場合は、プロセッサ プール メモリにフォール バックします。ファースト メモリは、ハードウェアの Interface Descriptor Block (IDB; インターフェイス デスクリプション ブロック) データ構造による使用のために確保された少量のメモリです。このメモリはブートアップ中も完全に予約され、メモリが完全に消費されているために常に高い使用率が示されます。このため、ファースト メモリ プールで空きメモリの低下が見られることは正常です。

%SNMP-4-HIGHCPU: プロセスは[DEC] ms しきい値 ([DEC] ms IOS クォンタム) をのための[文字]の超過します[文字]--結果[文字]

CPU ホグ メッセージは次のようになります。

```
SNMP-4-HIGHCPU: Process exceeds 200ms threshold (200ms IOS quantum)
for GET of rmon.19.16.0--result rmon.19.16.0
```

IOS の 12.4(13) では、新しいメッセージ (HIGHCPU) が追加されました。 CPU がプロセスにより 200 ミリ秒以上占有されると、HIGHCPU メッセージが報告されます。 HIGHCPU メッセージによるルータへの影響はありません。 どのプロセスが CPU の高使用率の原因であるのかが示されているだけです。 HIGHCPU メッセージは CPUHOG メッセージに似ていますが、HIGHCPU メッセージには大幅に低い許容しきい値が設定されています (CPUHOG メッセージと比較した場合、1/10 の時間 (ミリ秒での測定))。 2600 の 12.4(13) より前のバージョンでは、プロセスはより長い時間実行されましたが、この IOS バージョンにはこの機能拡張が用意されていなかったため、メッセージは生成されませんでした。

SNMP PDU プロセス (MIB オブジェクト クエリ) がサポートされ、PDU 内の各オブジェクトがほぼ同時に取得されるように、単一の CPU タイム クォンタムで実行されます。 これは SNMP プロトコル標準によって求められる要件です。 一部のオブジェクトはシステム内の大量のデータの集約であるため、これらが単一のオブジェクトであったとしても、その実装の方法により大量のプロセスが関与します。 MIB の実装ルールで要求されるように CPU が解放されない場合、このエラー メッセージが発生する可能性があります。 また、この同じ理由により、いくつかの異なるオブジェクトを同じオブジェクト グループまたはテーブル内でポーリングし、エラー メッセージが表示される場合は異常ではありません。

このメッセージがより多くの CPU タイムを使用する期待されるよりオブジェクトを識別するのに使用されています (まだない CPUHOG)。 一部の NMS または実装ツールは、ポーリング時に正常に動作しません。 この問題は Cisco バグ ID [CSCsl18139](#) ([登録ユーザのみ](#)) で文書化されています。

ソフトウェア暗号化による高CPU

デバイスにインストールされるハードウェア暗号化モジュールがない場合デバイスを通して来るすべての暗号化トラフィックはソフトウェアによって暗号化されなければなりません。 これは非常に CPU 中心です。 それは適度なスループット 要件とあらゆる暗号化配備のためにソフトウェア暗号化を使用するために推奨することではないです。 この問題を解決する 1 つのオプションは暗号化トラフィックの量を減らすことです (トラフィックを再ルーティングするか、または暗号化される) フローを制限して下さい。 ただし、この問題に対処する最もよい方法はハードウェア暗号化モジュールをソフトウェアによって起こる暗号化のための必要を省くこのデバイスのためにインストールされて得ることです。

注: トンネル/物理インターフェイスのクリプト マップの有効になることはメモリ消費 プロセスで、CPU の増加を引き起こす場合があります。

フラグメンテーションによる CPU使用率が高い状態

再組立ては CPU の上で CPU が多数のパケットを再構成しなければならない場合非常に高く駆動できます。

フラグメンテーションによる CPU使用率が高い状態のトラブルシューティングために [TCP を ms 調節します](#) TCP 同期する/ルータを通過する開始する (SYN) パケットの Maximum Segment Size (MSS) 値を設定する インターフェイスの [1400](#) コマンドを発行して下さい。

詳細情報を取得するためのコマンド

次のコマンドを使用することで、問題に関するより多くの情報を得ることができます。

- [show processes cpu](#)
- [show interfaces](#)
- [show interfaces switching](#)
- [show interfaces stat](#)
- [show ip nat translations](#)
- [show align](#)
- [show version](#)
- [show log](#)

ルータがまったくアクセス不能になった場合は、まず、ルータの電源を一旦オフにしてから再度オンにします。それから、定期的にメッセージが syslog サーバ ログオンする必要がある **show log** コマンドを除いてこのセクションのコマンドの出力を、集めて下さい。出力を収集する間隔は5分にします。この [UNIX シェル スクリプト](#) を使用すると、データを手動または自動で取得できます。また、HTTP または SNMP を使用して収集することも可能です。Cisco ルータで HTTP および SNMP を設定する方法については、ドキュメント『[Cisco IOS ソフトウェアの設定](#)』を参照してください。

show processes cpu コマンド

show processes cpu コマンドのヘッダー例を次に示します。

```
CPU utilization for five seconds: X%/Y%; one minute: Z%; five minutes: W%
PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
```

次の表に、ヘッダー内のフィールドの説明を示します。

フィールド説明

| | |
|---------|---------------------------------|
| X | 最後の 5 秒間の合計使用率の平均 (割り込みとプロセス) |
| Y | 最後の 5 秒間の、割り込みによる使用率の平均 |
| Z | 最後の 1 分間の合計使用率の平均 2 |
| W | 最後の 5 分間の合計使用率の平均 * |
| PID | プロセス ID |
| Runtime | プロセスが使用した CPU 時間 (ミリ秒) |
| Invoked | プロセスがコールされた回数 |
| uSecs | 各呼び出しの CPU 時間のマイクロ秒 |
| 5Sec | 最後の 5 秒間のタスクによる CPU 利用率 |
| 1Min | タスクによる最後の 1 分間の CPU 使用率 |
| 5Min | タスクによる最後の 5 分間の CPU 使用率 2 |
| TTY | プロセスを制御する端末 |
| Process | プロセスの名前 |

1 プロセス レベルでの CPU 使用率 = X - Y

2 値は算術平均ではなく、指数減衰平均です。そのため、新しい値ほど、算出される平均に大きな影響を与えます。

[show processes cpu](#) コマンドの詳細な説明については、このリンクを使用してください。

注: 合計 CPU 使用率を、ルータの packets 交換能力の余力を示す尺度としてとらえないでください。Cisco 7500 ルータでは、Versatile Interface Processor (VIP) と Route/Switch Processor (RSP; ルート スイッチ プロセッサ) は CPU 使用率を線形ではレポートしません。秒当たりの packets 交換能力の半数近くは、CPU 使用率が 90 ~ 95% になった後に処理されます。

show interfaces コマンド

[show interfaces](#) コマンドの詳細な説明は、コマンドリファレンスに記載されています。

show interfaces switching コマンド

このコマンドは、インターフェイス上のアクティブなスイッチングパスを確認する際に使用します。Cisco IOS ソフトウェアのスイッチングパスについての詳細は、ドキュメント『[スイッチングパスの設定](#)』を参照してください。

次に、インターフェイスが1つの場合の **show interfaces switching** コマンドの出力例を示します。

```
RouterA#show interfaces switching Ethernet0 Throttle count 0 Drops RP 0 SP 0 SPD Flushes Fast 0
SSE 0 SPD Aggress Fast 0 SPD Priority Inputs 0 Drops 0 Protocol Path Pkts In Chars In Pkts Out
Chars Out Other Process 0 0 595 35700 Cache misses 0 Fast 0 0 0 0 Auton/SSE 0 0 0 0 IP Process 4
456 4 456 Cache misses 0 Fast 0 0 0 0 Auton/SSE 0 0 0 0 IPX Process 0 0 2 120 Cache misses 0
Fast 0 0 0 0 Auton/SSE 0 0 0 0 Trans. Bridge Process 0 0 0 0 Cache misses 0 Fast 11 660 0 0
Auton/SSE 0 0 0 0 DEC MOP Process 0 0 10 770 Cache misses 0 Fast 0 0 0 0 Auton/SSE 0 0 0 0 ARP
Process 1 60 2 120 Cache misses 0 Fast 0 0 0 0 Auton/SSE 0 0 0 0 CDP Process 200 63700 100 31183
Cache misses 0 Fast 0 0 0 0 Auton/SSE 0 0 0 0
```

出力には、そのインターフェイスで設定されているすべてのプロトコルのスイッチングパスがリストされるため、ルータを通過しているトラフィックの種類と量が簡単にわかります。出力フィールドの説明を次の表に示します。

フィールド定義

| | |
|--------------|--|
| Process | 処理されたパケット。これらはルータ宛てのパケット、またはファーストスイッチングキャッシュ内にエントリがなかったパケットです。 |
| Cache misses | ファーストスイッチングキャッシュ内にエントリがなかったパケット。この宛先（またはポート）に設定されているファーストスイッチングのタイプによる）の最初のパケットはプロセスされます。発信インターフェイスでファーストスイッチングが明示的に無効にされていない限り、後続のパケットはすべてファーストスイッチングで交換されます。 |
| Fast | ファーストスイッチングで交換されたパケット。ファーストスイッチングはデフォルトで自律スイッチング、シリコンスイッチング、または分散スイッチングによって交換されたパケットです。 |
| Auton/SSE | 自律スイッチング、シリコンスイッチング、または分散スイッチングによって交換されたパケット。スイッチプロセッサまたはシリコンスイッチプロセッサ（それぞれ、自律スイッチングまたはシリコンスイッチング用）を備えた Cisco 7000 シリーズルータ、または VIP（分散スイッチング用）を備えた Cisco 7500 シリーズルータでのみ利用可能です。 |

show interfaces stat コマンド

このコマンドは、**show interfaces switching** コマンドの要約バージョンです。インターフェイスが1つの場合の出力例を次に示します。

```
RouterA#show interfaces stat Ethernet0 Switching path Pkts In Chars In Pkts Out Chars Out
Processor 52077 12245489 24646 3170041 Route cache 0 0 0 0 Distributed cache 0 0 0 0 Total 52077
12245489 24646 3170041
```

show interfaces stat コマンドの出力は、使用可能な[設定済みスイッチングパス](#)に応じて、プラットフォームごとに異なります。

show ip nat translations

show ip nat translations コマンドは、ルータ上でアクティブになっている Network Address

Translation (NAT; ネットワーク アドレス変換) 変換を表示します。アクティブな変換は、それぞれ CPU 割り込みを生成し、ルータの CPU 使用率の合計に影響を与えます。非常に多数の変換が行われると、ルータのパフォーマンスに影響があります。

次に、**show ip nat translations** コマンドの出力例を示します。

```
router#show ip nat translations Pro Inside global Inside local Outside local Outside global
--- 172.16.131.1 10.10.10.1 --- ---
```

show align コマンド

このコマンドは、Reduced Instruction Set Computing (RISC) プロセッサベースのプラットフォームでのみ使用できます。これらのプラットフォームの CPU には、アラインメントが誤っているメモリの読み取りまたは書き込みを訂正する機能があります。次に、出力例を示します。

```
Alignment data for:
4500 Software (C4500-DS40-M), Version mis-aligned RELEASE SOFTWARE (fcl)
Compiled Tue 31-Mar-98 15:05 by jdoe
```

```
Total Corrections 33911, Recorded 2, Reads 33911, Writes 0
```

```
Initial Initial
Address Count Access Type Traceback
40025F4D 15561 16bit read 0x606F4A7C 0x601C78F8 0x6012FE94 0x600102C0
40025F72 18350 32bit read 0x606FB260 0x6013113C 0x600102C0 0x60010988
```

show version コマンド

CPU 高負荷の問題を追跡するという目的からすると、このコマンド出力の重要な部分は、Cisco IOS ソフトウェアのバージョン、プラットフォーム、CPU のタイプ、およびルータの稼働時間です。[show version](#) コマンドの詳細な説明は、コマンド リファレンスに記載されています。

show log コマンド

このコマンドは、バッファに格納されているログ メッセージの内容を表示します。システム メッセージのロギングについての詳細は、『[ルータのトラブルシューティング](#)』設定ガイドの「ログ システムのエラー メッセージ」セクションを参照してください。

定期的にデータを収集するための UNIX シェル スクリプト

この付録では、一定の間隔でルータからデータをキャプチャする簡単なスクリプトについて説明します。このスクリプトの最も重要な部分は次の行です。

```
(echo "show version") | telnet 192.168.1.1
```

カッコ内のコマンドがサブシェルで実行され、その出力が Telnet セッションに送られます。**show version** コマンドと **show processes cpu** コマンドからの出力をキャプチャするスクリプト例を次に示します。

```
#!/opt/local/bin/bash

#####
# Router's IP address
#
IP_ADDRESS='10.200.40.53'
```



```

# Directory where the log files will be stored
#
DIR=/var/log/router

#####

if [ ! -e $DIR ]
then
  mkdir $DIR
fi

# Tag specification: mmddhhmm
DATE=`date +%m%d`
TIME=`date +%H%M`
TAG=$DATE$TIME

# Collect data from the router
(echo "foo";\
echo "bar";\
echo "term len 0";\
echo "show version";\
echo "show processes cpu";\
echo "term len 15";\
echo "show memory summary";\
echo "q";\
sleep 30)|telnet $IP_ADDRESS > $DIR/info.$TAG 2>$DIR/info.$TAG.msg

```

注: このスクリプトでは、パスワードも含めたすべてのデータがクリアテキスト形式で送信されま
す。

最初のセクションでは、IP アドレスと、ログ ファイルの保存先ディレクトリを指定する必要があります。2 番目のセクションには、ルータに実際に送信するコマンドを記述します。最初がユー
ザ名、次がパスワード、以降にコマンドを続けます。ここでは、特定のコマンドの最初の出力行
のみをキャプチャするためのテクニックが使用されています。プロンプトを介して、端末の長さ
を短めに設定し (このケースでは 15)、「q」文字を送信しています。

一定の間隔でデータを収集すると、`show version` の出力から、問題が周期的に発生する性質を持
っているか (たとえば、常に特定の時間帯や曜日に発生するなど) がわかります。さらに多くの
コマンドの出力を収集する場合は、上記の例と同じ方法でそれらのコマンドをスクリプトに追加
します。ファイルに送信する出力を切り捨てる場合は、最初にカッコ内に `sleep` コマンドを指定
して、スリープ時間を長くします。

CPU 高使用率の問題がたびたび発生するにもかかわらず、短時間で終結する場合は、このスクリ
プトを 5 分間隔で実行します。そうでない場合は、15 分または 30 分間隔で実行します。簡単
に使用するために、このスクリプトを `/usr/bin/router-script` などのファイルとして保存しておきま
す。さらに、5 分間隔で実行するには、`/etc/crontab` ファイルに次の行を追加します。

```
* /5 * * * * /usr/bin/router-script
```

cron サーバを再起動します。 `/etc/crontab` ファイルを変更する権限がない場合は、次のように独
立したプロセスとしてスクリプトを実行します。

```
while [ 1 ]; do ./router-script ; sleep 300; done &
```

関連情報

- [show processes コマンド](#)
- [Catalyst 2900XL/3500XL スイッチでの CPU の高使用率](#)
- [パフォーマンス チューニングの基本](#)

- [テクニカルサポートとドキュメント - Cisco Systems](#)