

SNMPプロセスが原因でCPU使用率が高い場合のトラブルシューティング

内容

[はじめに](#)

[使用するコンポーネント](#)

[ログ収集](#)

[ログ分析](#)

[SNMPビューの設定](#)

[トラブルシューティング情報](#)

[結論](#)

[関連情報](#)

はじめに

このドキュメントでは、9800ワイヤレスLANコントローラ(WLC)のSNMPプロセスに対する高いCPU使用率をトラブルシューティングし、解決するための構造化されたアプローチについて説明します。

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- ワイヤレスコントローラ：17.09.03を実行するC9800-80-K9

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

ログ収集

CPU使用率パターンの特定SNMPプロセスにリンクされているCPU使用率が高いというレポートを受け取った場合、最初の手順は、指定されたタイムフレームにわたって詳細なログを収集することです。これは、SNMPプロセスが最もアクティブでリソースを大量に消費する時間を特定するために不可欠な、CPU使用率のパターンまたは傾向を確立するのに役立ちます。

ログ収集を開始する前に、トラブルシューティングプロセスをサポートするために使用される特定の情報を収集することが不可欠です。まず、この問題に関する情報をいくつか収集します。

- システムでスパイクが発生していないか、使用率が一貫して高いか。
- どちらの場合も、使用率は何パーセントですか。

- CPU使用率が高い頻度は何ですか。
- 各SNMPサーバがWLCをポーリングする頻度はどのくらいですか。
- トップトーカーは誰ですか。

10分のスパンで2分間隔で9800 WLCからのコマンド出力を収集します。このデータを使用して、高いCPU使用率の問題、特にSNMPプロセスに関連する問題を分析できます。

```
#terminal length 0
#show clock
#show process cpu sorted | exclude 0.0
#show process cpu history
#show processes cpu platform sorted | exclude 0.0
#show snmp stats oid
#show snmp stats hosts
```

ログ分析

これらのログを収集した後、影響を理解するために分析する必要があります。

CPU使用率ログの例を見て、最もCPUを消費しているSNMPプロセスを特定してみましょう。

<#root>

```
WLC#show process cpu sorted | exclude 0.0
CPU utilization for five seconds: 96%/7%; one minute: 76%; five minutes: 61%
PID Runtime(ms)      Invoked      uSecs   5Sec   1Min   5Min TTY Process
 250  621290375      58215467      10672
58.34% 39.84% 34.11%    0 SNMP LA Cache pr  <-- High utilization

 93   167960640      401289855          418 14.50% 11.88%  9.23%  0 IOSD ipc task
739   141604259      102242639         1384  8.57%  6.95%  7.21%  0 SNMP ENGINE
763     7752         34896           222  4.00%  3.41%  1.83%  5 SSH Process
648   6216707       181047548           34  0.72%  0.37%  0.31%  0 IP SNMP
376   3439332       51690423            66  0.40%  0.36%  0.25%  0 SNMP Timers
143   3855538       107654825            35  0.40%  0.35%  0.23%  0 IOSXE-RP Punt Se
108   6139618       17345934            353  0.40%  0.30%  0.34%  0 DBAL EVENTS
```

show process cpu sorted | exclude 0.0コマンドの出力は、SNMPプロセスが実際に不均衡な量のCPUリソースを消費していることを示しています。具体的には、SNMP LA Cache prプロセスが最もCPUに負荷がかかり、その後他のSNMP関連プロセスが続きます。

次の一連のコマンドは、SNMPの使用率が高いプロセスを掘り下げる際に役立ちます。

<#root>

```
WLC#show snmp stats oid
```

```

time-stamp          #of times requested          OID
11:02:33 Austral Jun 8 2023

27698              bsnAPIfDBNoisePower <-- Frequently polled OID

11:02:23 Austral Jun 8 2023      1          sysUpTime
11:02:23 Austral Jun 8 2023     17         cLSiD11SpectrumIntelligenceEnable
11:02:23 Austral Jun 8 2023      1         cLSiD11SpectrumIntelligenceEnable
11:02:23 Austral Jun 8 2023      6         cLSiD11Band
11:02:23 Austral Jun 8 2023      1         cLSiD11Band
11:02:23 Austral Jun 8 2023      1         cLSiD11Band
11:02:23 Austral Jun 8 2023      1         cLSiD11Band
11:02:19 Austral Jun 8 2023     24         cTcCdpApCacheApName
11:02:19 Austral Jun 8 2023      1         cTcCdpApCacheDeviceIndex
11:02:19 Austral Jun 8 2023      9         cLApCpuAverageUsage
11:02:19 Austral Jun 8 2023    1315      cLApCpuCurrentUsage
11:02:19 Austral Jun 8 2023    2550      bsnAPIfDBNoisePower

```

show snmp stats oidコマンドの出力は、さまざまなOIDがポーリングされている頻度を明らかにします。bsnAPIfDBNoisePowerという特定のOIDは、要求数が非常に多いため、際立っています。これは、このOIDの積極的なポーリングが、WLCで見られる高いCPU使用率の一因である可能性があることを示唆しています。

OIDのbsnAPIfDBNoisePowerの動作と、そのデータの格納時間を理解してみましょう。

[SNMPオブジェクトナビゲータ](#)に移動し、OIDの「bsnAPIfDBNoisePower」を検索します。

Translate OID into object name or object name into OID to receive object details

Enter OID or object name: examples -
 OID: 1.3.6.1.4.1.9.9.27
Object Name: ifindex

Object Information

| Specific Object Information | |
|-----------------------------|--|
| Object | bsnAPIfDBNoisePower |
| OID | 1.3.6.1.4.1.14179.2.2.15.1.21 |
| Type | Integer32 |
| Permission | read-only |
| Status | current |
| MIB | AIRESpace-WIRELESS-MIB ; - View Supporting Images |
| Description | "This is the average noise power in dBm on each channel that is available to Airespace AP" |

OIDの検索結果

これで、bsnAPIfDBNoisePowerオブジェクトが、各APによって報告された各チャンネルのノイズ

出力を報告することについて理解できました。WLCによって管理されるチャンネルとAPの数が多いことを考えると、このOIDによって生成されるSNMPデータは大量になる可能性があります。WLCが多数のAPにサービスを提供する場合、このOIDのポーリングによって生成されるデータの量は膨大になる可能性があります。これは、WLCがこれらの大量のSNMP要求を処理するため、高いCPU使用率につながる可能性があります。

同様に、積極的にポーリングされる特定のOIDの動作を理解する必要があります。

次のコマンドは、WLCをポーリングしているSNMPサーバを確認するのに役立ちます。

```
WLC#show snmp stats hosts
Request Count    Last Timestamp  Address
77888844        00:00:00 ago    10.10.10.120
330242          00:00:08 ago    10.10.10.150
27930314        00:00:09 ago    10.10.10.130
839999          00:00:36 ago    10.10.10.170
6754377         19:45:34 ago    10.10.10.157
722             22:00:20 ago    10.10.10.11
```

このコマンドは、SNMPサーバのリストと、その要求カウントおよびポーリングアクティビティの最後のタイムスタンプを表示します。

9800 WLCをポーリングしている複数の異なるサーバがあることがわかります。過去10分間に収集された完全なログデータを確認すると、ポーリング頻度も測定できます。

これで、各サーバに移動して、問題のOIDがポーリングされる頻度を確認できます。この例では、OIDが30秒ごとにポーリングされており、必要以上に頻繁にポーリングされています。WLCは180秒ごとにRF/RRMデータを受信するため、OIDを30秒ごとにポーリングすると不要な処理が発生し、CPU使用率が高くなります。

問題のOIDとサーバが特定されたら、複数の異なるソリューションを試してWLCの負荷を軽減できます。

1. SNMPサーバでのポーリング頻度を減らす。
2. 操作の使用にOIDが必要ない場合は、SNMPサーバからのOIDのポーリングを無効にします。
3. SNMPサーバを制御できない場合は、SNMPビューを使用して問題のOIDをブロックできます。

SNMPビューの設定

ブロックするOIDを除外する新しいビューを定義します。たとえば、OID 1.3.6.1.4.1.14179.2.2.15.1.21をブロックして新しいビューを作成し、そのビューにOIDをアタッチするとします。

<#root>

```
snmp-server view blockOIDView
1.3.6.1.4.1.14179.2.2.15.1.21
excluded
<-- This is the OID of bsnAPIfDBNoisePower

snmp-server community TAC view blockOIDView RO
<-- This command assigns the blockOIDView to the community myCommunity with read-only (RO) access.

snmp-server group TAC v3 priv read blockOIDView
<-- This command assigns the blockOIDView to the group myGroup with the priv security level for SNMPv3.
```

トラブルシューティングのヒント

- CPU使用率のベースライン：SNMPプロセスによって高い使用率が生じていない場合の、通常のCPU使用率レベルを文書化します。
- SNMP設定：コミュニティストリング、バージョン（v2cまたはv3）、アクセスリストなど、現在のSNMP設定を確認します。
- SNMPのベストプラクティス：9800 WLCのベストプラクティスドキュメントを使用し、SNMPの推奨設定をできる限り近くに一致させてください。

```
C9800(config)#snmp-server subagent cache
C9800(config)#snmp-server subagent cache timeout ?
<1-100> cache timeout interval (default 60 seconds)
```

- SNMPポーリングの頻度：頻度を高くするとCPU負荷の増加につながるため、SNMPクエリによってWLCがポーリングされる頻度を決定します。
- ネットワークトポロジとSNMPマネージャ：ネットワーク設定を理解し、WLCと対話しているすべてのSNMPマネージャを特定します。
- System Uptime:最後のリポートからの経過時間をチェックして、稼働時間とCPU使用率に相関関係があるかどうかを確認します。
- 最近の変更：WLCの設定またはネットワークに対する最近の変更で、CPU使用率が高くなる時期と一致する可能性のあるものに注目してください。
- 9800 WLCでは、テレメトリに重点が置かれています。テレメトリは「プッシュ」モデルで動作します。このモデルでは、WLCはクエリを実行することなく、関連する情報をサーバに送信します。SNMPクエリがWLCのCPUサイクルを消費し、操作の問題を引き起こしている場合は、テレメトリに移行することをお勧めします。

結論

CPU使用率データを系統的に分析し、それをSNMPポーリングアクティビティと関連付けることで、Cisco 9800 WLC上のSNMPプロセスによって発生するCPU高使用率の問題をトラブルシュー

ティングし、解決できます。実装後のモニタリングは、トラブルシューティング作業の成功を確認し、最適なネットワークパフォーマンスを維持するために不可欠です。

関連情報

- [OIDによるSNMP経由のCatalyst 9800 WLCのモニタ](#)
- [SNMP V2、V3、およびNetCONFを使用したPrime InfrastructureによるCatalyst 9800ワイヤレスコントローラシリーズの管理](#)
- [Simple Network Management Protocol の保護](#)
- [Cisco Catalyst 9800シリーズの設定のベストプラクティス](#)

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。