

# ASR 1000 システム使用率をモニタする SNMP オブジェクト識別子

## 目次

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[Cisco IOSd メモリ使用をモニタする SNMP OID](#)

[RP/ESP/SIP CPU稼働率を監視する SNMP OID](#)

[RP/ESP/SIP メモリ使用をモニタする SNMP OID](#)

[イネーブル CoPP SNMP Overpolling から保護するため](#)

## 概要

この文書は Cisco ASR 1000 シリーズ モジュラールータの CPU およびメモリ リソースを監視するために使用されるべき推奨されるオブジェクト識別子 (OID) を説明したものです。ソフトウェアベースの転送プラットフォームとは違って、ASR 1000 シリーズはシステムのこれらの機能要素から成り立ちます:

- ASR 1000 シリーズ ルート プロセッサ (RP)
- ASR 1000 シリーズ エンベデッド サービス プロセッサ (ESP)
- ASR 1000 シリーズ SPA インターフェイス プロセッサ (SIP)

そのように、管理対象装置ごとにポーリングされるべき追加 OID という結果に終る実稼働環境のこれらのプロセッサのそれぞれによって CPU およびメモリ使用をモニタすることを必要とします。

## 前提条件

### 要件

次の項目に関する知識が推奨されます。

- Simple Network Management Protocol (SNMP)
- Cisco IOS<sup>®</sup>-XE

### 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。

本書の情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。このドキュメントで使用するすべてのデバイスは、初期 (デフォルト) 設定の状態から起動しています。稼働中のネットワークで作業を行う場合、コマンドの影響について十分に理解したうえで作業してくだ

さい。

## Cisco IOSd メモリ使用をモニタする SNMP OID

ASR 1000 で、メモリ使用量を監視するために設計されている 64 ビット アーキテクチャ プラットフォームのために OID を使用する必要があります:

プロセッサ プール 空きメモリ	1.3.6.1.4.1.9.9.221.1.1.1.1.20.7000.1 ( MIBcempMemPoolHCFree )
プロセッサ プール最も大きいメモリ	1.3.6.1.4.1.9.9.221.1.1.1.1.22.7000.1 ( MIBcempMemPoolHCLargestFree )
プロセッサ プールによって使用されるメモリ	1.3.6.1.4.1.9.9.221.1.1.1.1.18.7000.1 ( MIBcempMemPoolHCUsed )
プロセッサ プール メモリ不足	1.3.6.1.4.1.9.9.221.1.1.1.1.24.7000.1 ( MIBcempMemPoolHCLowestFree )

注: Cisco IOSd メモリ統計情報をポーリングするためにより少ない仕様 OID を使用する場合はシステムは 2 出力を- Cisco IOSd 空きメモリ ( OID-7000.1 ) および Linux 共用メモリ パート インターフェイス ( LSMPI ) メモリ ( OID-7000.2 ) もたらしめます。これにより管理ステーションは LSMPI プールのためのメモリ不足 アラートを報告しますかもしれません。LSMPI メモリプールは Forwarding Processor からルートプロセッサにパケットを転送するために使用されます。ASR 1000 プラットフォームで、lsmpi\_io プールに正常である少し空きメモリが- 1000 バイト一般に以下あります。シスコは、誤ったアラームを回避するためにネットワーク管理アプリケーションによる LSMPI プールのモニタリングを無効にすることを推奨します。

## RP/ESP/SIP CPU稼働率を監視する SNMP OID

```
ASR1K#show platform software status control-processor brief | section Load
```

```
Load Average
Slot      Status      1-Min   5-Min   15-Min
RP0       Healthy     0.75    0.47    0.41
ESP0      Healthy     0.00    0.00    0.00
SIP0      Healthy     0.00    0.00    0.00
```

それに対応します:

```
ASR1K#show platform software status control-processor brief | section Load
```

```
Load Average
Slot      Status      1-Min   5-Min   15-Min
RP0       Healthy     0.75    0.47    0.41
ESP0      Healthy     0.00    0.00    0.00
SIP0      Healthy     0.00    0.00    0.00
```

上の OID を使用方法を ASR 1000 カーネル ロード CPU を監察するために説明する [EEM スクリプトとの ASR カーネル ロード CPU のモニタ](#)を参照して下さい。

注: RP2 は 2 つの物理的な CPU が含まれていますが、CPU は別々に監視されません。CPU稼働率は両方の CPU の集約結果であり、従って cpmCPUTotalTable オブジェクトは RP CPU のための 1 つのエントリだけ含まれています。これにより時折管理ステーションは 100% の上の CPU稼働率を報告しますかもしれません。

# RP/ESP/SIP メモリ使用をモニタする SNMP OID

これらの出力は提示プラットフォームソフトウェア ステータス Control Processor 要約コマンドによって感知される各プロセッサの個々のメモリ統計情報をポーリングするために OID をリストします。

```
ASR1K#show platform software status control-processor brief | s Memory
Memory (kB)
Slot   Status   Total           Used(Pct)       Free (Pct)      Committed (Pct)
RP0    Healthy  3874504         2188404 (56%)  1686100 (44%)  2155996 (56%)
ESP0   Healthy  969088          590880 (61%)   378208 (39%)   363840 (38%)
SIP0   Healthy  471832          295292 (63%)   176540 (37%)   288540 (61%)
```

```
ASR1K#show platform software status control-processor brief | s Memory
Memory (kB)
Slot   Status   Total           Used(Pct)       Free (Pct)      Committed (Pct)
RP0    Healthy  3874504         2188404 (56%)  1686100 (44%)  2155996 (56%)
ESP0   Healthy  969088          590880 (61%)   378208 (39%)   363840 (38%)
SIP0   Healthy  471832          295292 (63%)   176540 (37%)   288540 (61%)
```

注: 前の OID は ASR 1001 および ADR 1002-X のような 1RU (ラックマウント単位) プラットフォームのための単一出力だけでもたらしめます。ASR 1001 の制御 CPU に 3 つの論理関数が - RP、FP (Forwarding Processor)、および CC (キャリアカード) あります。ASR 1002 の異なる委員会を渡って普通広がるすべての機能は ASR 1001 の同じ CPU で動作します。

## イネーブル CoPP SNMP Overpolling から保護するため

コントロールプレーン ポリシング (CoPP) の設定はサービス拒絶 (DoS) 攻撃の場合によりよいプラットフォーム信頼性およびアベイラビリティを提供します。CoPP 機能は入力および出トラフィックのための自身のインターフェイスと別途のエンティティとしてコントロールプレーンを扱います。このインターフェイスはパントと呼ばれました/インターフェイスをインジェクトします。段階的に行なわれたアプローチでされる CoPP ポリシー必要の配備。最初のフェーズは寛大な状態でテストの分析を可能にするために最初の移行/ディプロイメントフェーズ パケットのポリシングを行ない。展開されて、CoPP ポリシーと関連付けられるクラスのそれぞれはチェックされる調節する必要があり、レート。方法の代表的な例は overpolling からコントロールプレーンを保護することを CoPP が可能にするここに示されています:

```
ASR1K#show platform software status control-processor brief | s Memory
Memory (kB)
Slot   Status   Total           Used(Pct)       Free (Pct)      Committed (Pct)
RP0    Healthy  3874504         2188404 (56%)  1686100 (44%)  2155996 (56%)
ESP0   Healthy  969088          590880 (61%)   378208 (39%)   363840 (38%)
SIP0   Healthy  471832          295292 (63%)   176540 (37%)   288540 (61%)
```

ここに示されるように policy-map をアクティブにして下さい:

```
ASR1K#show platform software status control-processor brief | s Memory
Memory (kB)
Slot   Status   Total           Used(Pct)       Free (Pct)      Committed (Pct)
RP0    Healthy  3874504         2188404 (56%)  1686100 (44%)  2155996 (56%)
ESP0   Healthy  969088          590880 (61%)   378208 (39%)   363840 (38%)
SIP0   Healthy  471832          295292 (63%)   176540 (37%)   288540 (61%)
```