

ASR 1000 シリーズ ルータ メモリのトラブルシューティング ガイド

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[ASR メモリ レイアウトの概要](#)

[lsmapi io プールでのメモリの割り当て](#)

[メモリ使用量](#)

[IOS-XE でのメモリ使用量の検証](#)

[IOSd でのメモリ使用量の検証](#)

[ASR1K での TCAM 使用の検証](#)

[QFP でのメモリ使用量の検証](#)

概要

このドキュメントでは、システム メモリを確認して Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ (ASR1K) のメモリ関連の問題を修復する方法について説明します。

前提条件

要件

次の項目に関する基本的な知識が推奨されます。

- Cisco IOS XE ソフトウェア
- ASR CLI

注 : ASR 1001 シリーズ ルータの Linux シェルにログインするには、特別なライセンスが必要な場合があります。

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。

- すべての ASR1K プラットフォーム
- ASR1K プラットフォームをサポートするすべての Cisco IOS XE ソフトウェア リリース

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、初期 (デフォルト) 設定の状態から起動しています。対象のネットワークが稼働中である場合には、どのようなコマンドについても、その潜在的な影響について確実に理解しておく必要があります。

ASR メモリ レイアウトの概要

以前のソフトウェア ベースのルータ プラットフォームのほとんどでは、内部ソフトウェア プロセスの大部分は Cisco IOS[®] (IOS) メモリを使用して実行されます。ASR1K プラットフォームでは、オペレーティング システム (OS) の多数の役割を IOS プロセス外に移動する分散ソフトウェア アーキテクチャを導入しています。このアーキテクチャの iOS は、以前はほとんどすべての内部オペレーションを処理していましたが、多数の Linux プロセスの 1 つとして稼働するようになりました。これにより、他の Linux プロセスでルータを実行する役割を共有できるようになります。

ASR1K は、従来の IOS ではなく IOS XE を実行します。IOS XE では、Linux コンポーネントはカーネルを実行し、IOS はデーモンとして稼働するため、今後は IOSd (IOS デーモン) と呼ばれます。これにより、メモリを Linux カーネルと IOSd インスタンス間でメモリを分割する必要があります。

IOSd と残りのシステム間で分割されるメモリは起動時に固定されており、変更できません。4 GB システムでは、IOSd に約 2 GB 割り当てられ、8 GB のシステムでは、IOSd に約 4 GB 割り当てられます (ソフトウェア冗長性は無効の状態)。

ASR1K には 64 ビット アーキテクチャがあるため、システムのすべてのデータ構造にあるポインタは、従来のシングル CPU プラットフォームと比べて倍の量のメモリを消費します (4 バイトではなく、8 バイト)。64 ビット アドレッシングにより、IOS では IOS の 2 GB アドレス可能なメモリ制限を克服することができるため、数百万のルートまで拡張することができます。

注：新しい機能をアクティブにする前に、使用可能なメモリが十分あることを確認します。シスコは、メモリ不足を防ぐためにソフトウェア冗長性を有効にする際にボーダー ゲートウェイ プロトコル (BGP) 全体のルーティング テーブルを受信する場合、最低 8 GB DRAM があることを推奨します。

lsmpi_io プールでのメモリの割り当て

Linux Shared Memory Punt Interface (LSMPI) のメモリ プールは、フォワーディング プロセッサからルート プロセッサにパケットを転送するために使用されます。プロセッサ プールとは反対に、このメモリ プールはルータの初期化時に事前に割り当てられたバッファに分割され、そこで IOS XE がメモリ ブロックを動的に割り当てます。ASR1K プラットフォームでは、lsmpi_io プールの空きメモリは一般に 1000 バイト未満であり、通常の状態です。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。

```
ASR1000# show memory statistics
```

	Head	Total(b)	Used(b)	Free(b)	Lowest(b)	Largest(b)
Processor	2C073008	1820510884	173985240	1646525644	1614827804	1646234064

```
lsmpi_io 996481D0 6295088 6294120 968 968 968
```

LSMPI パスに問題がある場合、次のように **Device xmit fail** のカウンタがこのコマンドの出力で増加します (一部の出力は省略)。

```
ASR1000-1# show platform software infrastructure lsmpi driver
```

```
LSMPI Driver stat ver: 3
```

```
Packets:
```

```
In: 674572
```

```
Out: 259861
```

```
Rings:
```

```
RX: 2047 free 0 in-use 2048 total
```

```
TX: 2047 free 0 in-use 2048 total
```

```
RXDONE: 2047 free 0 in-use 2048 total
```

```
TXDONE: 2047 free 0 in-use 2048 total
```

```
Buffers:
```

```
RX: 7721 free 473 in-use 8194 total
```

```
Reason for RX drops (sticky):
```

```
Ring full : 0
```

```
Ring put failed : 0
```

```
No free buffer : 0
```

```
Receive failed : 0
```

```
Packet too large : 0
```

```
Other inst buf : 0
```

```
Consecutive SOPs : 0
```

```
No SOP or EOP : 0
```

```
EOP but no SOP : 0
```

```
Particle overrun : 0
```

```
Bad particle ins : 0
```

```
Bad buf cond : 0
```

```
DS rd req failed : 0
```

```
HT rd req failed : 0
```

```
Reason for TX drops (sticky):
```

```
Bad packet len : 0
```

```
Bad buf len : 0
```

```
Bad ifindex : 0
```

```
No device : 0
```

```
No skbuff : 0
```

```
Device xmit fail : 0
```

```
Device xmit retry : 0
```

```
Tx Done ringfull : 0
```

```
Bad u->k xlation : 0
```

```
No extra skbuff : 0
```

```
<snip>
```

メモリ使用量

ASR1K はシステムの次の機能要素から構成されます。

- 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)

したがって、実稼動環境でこれらのプロセッサのメモリ使用量をモニタする必要があります。

コントロール プロセッサは、Linux ベースのカーネルと、RP カード上でユーザ プロセスとして稼働する Cisco IOS ソフトウェアを含む共通の OS レベルのユーティリティ プログラム セットで構成されている Cisco IOS-XE ソフトウェアを実行します。

IOS-XE でのメモリ使用量の検証

RP、ESP、SIP でメモリ使用量を監視するには、`show platform software status control-processor brief` コマンドを入力します。メモリ使用量を比較する際は、機能の設定とトラフィックなどの側面に関してシステム状態が同一である必要があります。

```
ASR1K# show platform software status control-processor brief
<snip>
```

```
Memory (kB)
Slot Status   Total      Used (Pct)   Free (Pct)  Committed (Pct)
RP0 Healthy 3907744    1835628 (47%) 2072116 (53%) 2614788 (67%)
ESP0 Healthy 2042668    789764 (39%) 1252904 (61%) 3108376 (152%)
SIP0 Healthy 482544     341004 (71%) 141540 (29%) 367956 (76%)
SIP1 Healthy 482544     315484 (65%) 167060 (35%) 312216 (65%)
```

注：コミットされたメモリは、システムがこの作業負荷に対して絶対にメモリ不足 (OOM) にならないことを保証するために必要なメモリ容量の推定値です。通常は、カーネルがメモリをオーバーコミットします。たとえば、1 GB のメモリを割り当てても、実際には何も起こりません。その割り当て済みメモリで、使用するだけのメモリの使用が開始されるまで、そのメモリは要求されません。

上の出力に表示される各プロセッサは、空きメモリの容量に応じて **Healthy**、**Warning**、または **Critical** ステータスを示す場合があります。プロセッサのいずれかが **Warning** または **Critical** のステータスを示した場合は、一番の原因を特定するために `monitor platform software process<slot>` コマンドを入力します。

```
ASR1K# monitor platform software process ?
0 SPA-Inter-Processor slot 0
1 SPA-Inter-Processor slot 1
F0 Embedded-Service-Processor slot 0
F1 Embedded-Service-Processor slot 1
FP Embedded-Service-Processor
R0 Route-Processor slot 0
R1 Route-Processor slot 1
RP Route-Processor
<cr>
```

`monitor platform software process` コマンドを実行する前に、以下のようにターミナルタイプの設定が要求される場合があります。

```
ASR1K# monitor platform software process r0
Terminal type 'network' unsupported for command
Change the terminal type with the 'terminal terminal-type' command.
```

デフォルトでは、ターミナルタイプは `network` に設定されています。適切なターミナルタイプを設定するには、`terminal terminal-type` コマンドを入力します。

```
ASR1K#terminal terminal-type vt100
```

正しいターミナルタイプを設定すると、`monitor platform software process` コマンドを入力することができます (一部の出力は省略)。

```
ASR1000# monitor platform software process r0
```

```
top - 00:34:59 up 5:02, 0 users, load average: 2.43, 1.52, 0.73
Tasks: 136 total, 4 running, 132 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.8%us, 2.3%sy, 0.0%ni, 96.8%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 2009852k total, 1811024k used, 198828k free, 135976k buffers
Swap: 0k total, 0k used, 0k free, 1133544k cached
```

```
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
25956 root 20 0 928m 441m 152m R 1.2 22.5 4:21.32 linux_iosd-imag
29074 root 20 0 106m 95m 6388 S 0.0 4.9 0:14.86 smand
24027 root 20 0 114m 61m 55m S 0.0 3.1 0:05.07 fman_rp
25227 root 20 0 27096 13m 12m S 0.0 0.7 0:04.35 imand
23174 root 20 0 33760 11m 9152 S 1.0 0.6 1:58.00 cmand
23489 root 20 0 23988 7372 4952 S 0.2 0.4 0:05.28 emd
24755 root 20 0 19708 6820 4472 S 1.0 0.3 3:39.33 hman
28475 root 20 0 20460 6448 4792 S 0.0 0.3 0:00.26 psd
27957 root 20 0 16688 5668 3300 S 0.0 0.3 0:00.18 plogd
14572 root 20 0 4576 2932 1308 S 0.0 0.1 0:02.37 reflector.sh
<snip>
```

注：メモリ使用量の降順で出力をソートするには、Shift + M キーを押します。

IOSd でのメモリ使用量の検証

linux_iosd-imag プロセスが monitor platform software process rp active コマンド出力できわめて大量のメモリを保持していることがわかった場合には、IOSd インスタンスのトラブルシューティングに取り組んでください。IOSd スレッドの特定のプロセスがメモリを解放していない可能性があります。Cisco 2800、3800、3900 シリーズなどのソフトウェアベースのフォワーディングプラットフォームのトラブルシューティングと同じ方法で IOSd インスタンスのメモリに関する問題のトラブルシューティングを行います。

```
ASR1K# monitor platform software process rp active
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
25794 root 20 0 2929m 1.9g 155m R 99.9 38.9 1415:11 linux_iosd-imag
23038 root 20 0 33848 13m 10m S 5.9 0.4 30:53.87 cmand
9599 root 20 0 2648 1152 884 R 2.0 0.0 0:00.01 top
<snip>
```

問題のあるプロセスを特定するには、show process memory sorted コマンドを入力します。

```
ASR1000# show process memory sorted
Processor Pool Total: 1733568032 Used: 1261854564 Free: 471713468
lsmpi_io Pool Total: 6295088 Used: 6294116 Free: 972

PID TTY Allocated Freed Holding Getbufs Retbufs Process
522 0 1587708188 803356800 724777608 54432 0 BGP Router
234 0 3834576340 2644349464 232401568 286163388 15876 IP RIB Update
0 0 263244344 36307492 215384208 0 0 *Init
```

注：メモリ使用量が適正であるかどうかについての特定とトラブルシューティングを行うためにサポートが必要な場合は、TAC ケースを開きます。

ASR1K での TCAM 使用の検証

トラフィック分類は、ルータとスイッチにある最も基本的な機能の 1 つです。多くのアプリケーションおよび機能では、インフラストラクチャ デバイスがこれらの差別化サービスを品質の要件に基づいた異なるユーザに提供することが必要です。トラフィック分類プロセスは、デバイスのスループットが著しく低下しないように迅速にする必要があります。このため、ASR1K プラットフォームでは、第 4 世代の Ternary Content Addressable Memory (TCAM4) を使用します。

プラットフォームで使用可能な TCAM セルの総数および残りの空きエントリ数を判断するには、次のコマンドを入力します。

```
ASR1000# show platform hardware qfp active tcam resource-manager usage
```

```
Total TCAM Cell Usage Information
```

```
-----  
Name : TCAM #0 on CPP #0  
Total number of regions : 3  
Total tcam used cell entries : 65528  
Total tcam free cell entries : 30422  
Threshold status : below critical limit
```

注：シスコでは、エントリをプログラムするために TCAM で利用可能な空きセルが十分保持するように、アクセスリストまたは Quality of Service (QoS) ポリシーを変更する前にしきい値のステータスを確認することを推奨します。

フォワーディング プロセッサが、TCAM の空きセルが非常に不足した状態で動作している場合、ESP は次のようなログを生成し、クラッシュすることがあります。冗長性がない場合は、トラフィックは中断します。

```
%CPPTCAMRM-6-TCAM_RSRC_ERR: SIP0: cpp_sp: Allocation failed because of insufficient TCAM resources in the system.
```

```
%CPPOS LIB-3-ERROR_NOTIFY: SIP0: cpp_sp:cpp_sp encountered an error -  
Traceback=1#s7f63914d8ef12b8456826243f3b60d7 errmsg:7EFC525C000+1175
```

QFP でのメモリ使用量の検証

物理メモリに加え、データ構造の転送に使用される Quantum Flow Processor (QFP) ASIC に割り当てられているメモリもあります。これには、転送情報ベース (FIB)、QoS ポリシーなどのデータが含まれています。QFP ASIC に使用できる DRAM の容量は、ESP モジュールに応じて 256 MB、512 MB、および 1 GB の範囲で固定されています。

exmem メモリ使用量を判断するには、`show platform hardware qfp active infrastructure exmem statistics` コマンドを入力します。使用される IRAM と DRAM のメモリの合計が使用中の合計 QFP メモリとなります。

```
BGL.I.05-ASR1000-1# show platform hardware qfp active infra exmem statistics user
```

```
Type: Name: IRAM, CPP: 0  
Allocations Bytes-Alloc Bytes-Total User-Name  
-----  
1 115200 115712 CPP_FIA  
Type: Name: DRAM, CPP: 0  
Allocations Bytes-Alloc Bytes-Total User-Name  
-----
```

4	1344	4096	P/I
9	270600	276480	CEF
1	1138256	1138688	QM RM
1	4194304	4194304	TCAM
1	65536	65536	Qm 16

IRAM は QFP ソフトウェアの命令メモリです。DRAM が足りなくなった場合には、利用可能な IRAM を使用できます。IRAM が、メモリが非常に不足した状態で動作している場合、次のエラーメッセージが表示されることがあります。

```
%QFPOOR-4-LOWRSRC_PERCENT: Fl: cpp_ha: QFP 0 IRAM resource low - 97 percent depleted
%QFPOOR-4-LOWRSRC_PERCENT: Fl: cpp_ha: QFP 0 IRAM resource low - 98 percent depleted
```

最も多くメモリを消費しているプロセスを判断するには、次のように **show platform hardware qfp active infra exmem statistics user** コマンドを入力します。

```
ASR1000# show platform hardware qfp active infra exmem statistics user
```

```
Type: Name: IRAM, CPP: 0
Allocations  Bytes-Alloc  Bytes-Total  User-Name
-----
1             115200       115712       CPP_FIA
```

```
Type: Name: DRAM, CPP: 0
Allocations  Bytes-Alloc  Bytes-Total  User-Name
-----
4             1344         4096         P/I
9             270600      276480      CEF
1             1138256    1138688    QM RM
1             4194304    4194304    TCAM
1             65536      65536      Qm 16
```