

ASR 1000 Series 라우터 메모리 문제 해결 가이드

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[ASR 메모리 레이아웃 개요](#)

[lsmpi io 풀 아래의 메모리 할당](#)

[메모리 사용량](#)

[IOS-XE에서 메모리 사용량 확인](#)

[IOSd에서 메모리 사용량 확인](#)

[ASR1K에서 TCAM 사용률 확인](#)

[QFP에서 메모리 사용률 확인](#)

소개

이 문서에서는 Cisco ASR 1000 Series Aggregation Services Router(ASR1K)에서 시스템 메모리를 확인하고 메모리 관련 문제를 해결하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

Cisco에서는 이러한 주제에 대한 기본적인 지식을 얻을 것을 권장합니다.

- Cisco IOS-XE 소프트웨어
- ASR CLI

참고:ASR 1001 Series 라우터의 Linux 셸에 로그인하려면 특별 라이선스가 필요할 수 있습니다.

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- 모든 ASR1K 플랫폼
- ASR1K 플랫폼을 지원하는 모든 Cisco IOS-XE 소프트웨어 릴리스

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우, 모든 명령어의 잠재적인 영향을 미리 숙지하시기 바랍니다.

ASR 메모리 레이아웃 개요

대부분의 소프트웨어 기반 라우터 플랫폼에서는 내부 소프트웨어 프로세스의 대부분이 Cisco IOS® 메모리 내에서 실행됩니다. ASR1K 플랫폼에는 IOS 프로세스에서 많은 OS(운영 체제) 책임을 이동하는 분산형 소프트웨어 아키텍처가 도입되었습니다. 대부분의 내부 작업을 담당했던 이 아키텍처의 IOS는 이제 많은 Linux 프로세스 중 하나로 실행됩니다. 이렇게 하면 다른 Linux 프로세스가 라우터 운영에 대한 책임을 공유할 수 있습니다.

ASR1K는 기존 IOS가 아니라 IOS-XE를 실행합니다. IOS-XE에서 Linux 구성 요소는 커널을 실행하며, IOS는 디먼으로 실행되며, 이후 IOSd(IOS-Daemon)라고도 합니다. 이렇게 하면 Linux 커널과 IOSd 인스턴스 간에 메모리를 분할해야 합니다.

IOSd와 나머지 시스템 간에 분할된 메모리는 시작 시 고정되며 수정할 수 없습니다. 4GB 시스템의 경우 IOSd는 약 2GB에 할당되며, 8GB 시스템의 경우 IOSd에 약 4GB가 할당됩니다(소프트웨어 리던던시가 비활성화됨).

ASR1K에는 64비트 아키텍처가 있으므로 시스템의 모든 데이터 구조에 있는 모든 포인터는 기존 단일 CPU 플랫폼(4바이트 대신 8바이트)과 비교할 때 2배의 메모리를 사용합니다. 64비트 주소 지정을 통해 IOS는 2GB 주소 지정 가능 메모리 제한을 극복하여 수백만 개의 경로로 확장할 수 있습니다.

참고: 새 기능을 활성화하기 전에 사용 가능한 메모리가 충분한지 확인합니다. 메모리 소모를 방지하기 위해 소프트웨어 이중화가 활성화된 경우 전체 BGP(Border Gateway Protocol) 라우팅 테이블을 수신하는 경우 8GB 이상의 DRAM이 있는 것이 좋습니다.

lsmpi_io 풀 아래의 메모리 할당

Linux LSMPI(Shared Memory Punt Interface) 메모리 풀은 포워딩 프로세서에서 경로 프로세서로 패킷을 전송하는 데 사용됩니다. 이 메모리 풀은 라우터 초기화 시 IOS-XE가 메모리 블록을 동적으로 할당하는 프로세서 풀과 달리 미리 할당된 버퍼에 할당됩니다. ASR1K 플랫폼에서 lsmpi_io 풀은 일반적으로 1000바이트 미만의 사용 가능한 메모리가 거의 없습니다. Cisco에서는 잘못된 경보를 방지하기 위해 네트워크 관리 애플리케이션에서 LSMPI 풀 모니터링을 비활성화할 것을 권장합니다.

```
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 경로에 문제가 있는 경우 **Device xmit fail** 카운터가 이 명령 출력에 증분됩니다(일부 출력 생략).

```

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 rtry : 0
    Tx Done ringfull : 0
    Bad u->k xlation : 0
    No extra skbuff  : 0
<snip>

```

메모리 사용량

ASR1K는 시스템의 다음과 같은 기능 요소로 구성됩니다.

- ASR 1000 Series RP(Route Processor)
- ASR 1000 Series ESP(Embedded Services Processor)
- ASR 1000 Series SIP(SPA Interface Processor)

따라서 프로덕션 환경에서 각 프로세서의 메모리 사용률을 모니터링해야 합니다.

제어 프로세서는 Linux 기반 커널과 OS 레벨 유틸리티 프로그램의 공통 집합으로 구성된 Cisco IOS-XE 소프트웨어를 실행합니다. 여기에는 RP 카드에서 사용자 프로세스로 실행되는 Cisco IOS가 포함됩니다.

IOS-XE에서 메모리 사용량 확인

RP, ESP 및 SIP의 메모리 사용률을 모니터링하려면 `show platform software status control-`

processor brief 명령을 입력합니다.시스템 상태는 기능 컨피그레이션 및 트래픽과 같은 측면과 메모리 사용량을 비교할 때 동일해야 합니다.

```
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%)
```

참고:Committed 메모리는 시스템이 이 워크로드에 대해 OOM(Out of Memory)이 되지 않도록 보장하기 위해 필요한 RAM의 양을 추정한 것입니다.일반적으로 커널은 메모리를 오버커밋합니다.예를 들어, 1GB malloc를 실행할 때 아무 일도 일어나지 않습니다.할당된 메모리를 사용하기 시작할 때만 실제 메모리 온디맨드 방식으로 수신되며 사용한 만큼만 사용할 수 있습니다.

이전 출력에 나열된 각 프로세서는 **Healthy**, **Warning** 또는 **Critical**로 상태를 보고할 수 있으며 이는 사용 가능한 메모리 양에 따라 달라집니다.프로세서 중 하나라도 **경고** 또는 **위험**으로 상태를 표시할 경우 상위 참가자를 식별하려면 `monitor platform software process<slot>` 명령을 입력합니다.

```
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` 명령을 실행하기 전에 `terminal-type`을 설정하라는 메시지가 표시될 수 있습니다.

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

터미널 유형은 기본적으로 **네트워크**로 설정됩니다.적절한 터미널 유형을 설정하려면 `terminal terminal-type` 명령을 입력합니다.

ASR1K#터미널 유형 vt100

올바른 터미널 유형이 구성되면 `monitor platform software process` 명령을 입력할 수 있습니다(일부 출력 생략).

```
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>

```

참고:출력을 메모리 사용량 내림차순으로 정렬하려면 **Shift + M**을 누릅니다.

IOSd에서 메모리 사용량 확인

linux_iosd-imag 프로세스가 모니터 플랫폼 소프트웨어 프로세스 rp active 명령 출력에서 비정상적으로 많은 양의 메모리를 보유한다는 사실을 알게 되면 IOSd 인스턴스에 대한 문제 해결 노력을 집중합니다.IOSd 스레드의 특정 프로세스가 메모리를 해제하지 않을 수 있습니다.Cisco 2800, 3800 또는 3900 Series와 같은 소프트웨어 기반 포워딩 플랫폼 문제를 해결하는 것과 동일한 방식으로 IOSd 인스턴스에서 메모리 관련 문제를 해결합니다.

```

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>

```

문제 프로세스를 식별하려면 **show process memory sorted** 명령을 입력합니다.

```

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

```

참고:트러블슈팅을 수행하거나 메모리 사용이 정당한지 확인하기 위해 도움이 필요한 경우 TAC 케이스를 엽니다.

ASR1K에서 TCAM 사용률 확인

트래픽 분류는 라우터와 스위치에 있는 가장 기본적인 기능 중 하나입니다.많은 애플리케이션과 기능을 사용하려면 인프라 장치가 품질 요구 사항에 따라 각기 다른 사용자에게 이러한 차별화된 서비스를 제공해야 합니다.디바이스의 처리량이 크게 저하되지 않도록 트래픽 분류 프로세스를 빠르게 수행해야 합니다.ASR1K 플랫폼은 4세대 TCAM4(Ternary Content Addressable Memory)를 이

한 용도로 사용합니다.

플랫폼에서 사용 가능한 TCAM 셀의 총 수 및 남은 사용 가능한 항목 수를 확인하려면 다음 명령을 입력합니다.

```
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
```

참고: 액세스 목록 또는 QoS(Quality of Service) 정책을 변경하기 전에 항상 임계값 상태를 확인하는 것이 좋습니다. 그러면 TCAM에 항목을 프로그래밍하기 위해 사용 가능한 셀이 충분히 있을 수 있습니다.

포워딩 프로세서가 사용 가능한 TCAM 셀에서 매우 낮은 속도로 실행되는 경우 ESP는 아래에 표시된 것과 유사한 로그를 생성하여 충돌할 수 있습니다. 이중화가 없으면 트래픽 중단이 발생합니다.

```
%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
```

QFP에서 메모리 사용률 확인

물리적 메모리 외에도 FIB(Forwarding Information Base) 및 QoS 정책과 같은 데이터를 포함하는 데이터 구조를 전달하는 데 사용되는 QFP(Quantum Flow Processor) ASIC에 연결된 메모리도 있습니다. QFP ASIC에 사용할 수 있는 DRAM의 양은 고정되어 있으며 범위는 256MB, 512MB 및 1GB이며, ESP 모듈에 따라 다릅니다.

exmem 메모리 사용량을 확인하려면 **show platform hardware qfp active infrastructure exmem statistics** 명령을 입력합니다. 사용되는 IRAM 및 DRAM의 메모리 합계는 사용 중인 총 QFP 메모리를 제공합니다.

```
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
```

IRAM은 QFP 소프트웨어의 명령 메모리입니다. DRAM이 모두 소모되면 사용 가능한 IRAM을 사용할 수 있습니다. IRAM이 메모리에서 매우 낮게 실행되는 경우 다음 오류 메시지가 표시될 수 있습니다.

```
%QFPOOR-4-LOWRSRC_PERCENT: F1: cpp_ha: QFP 0 IRAM resource low - 97 percent depleted
```

```
%QFPOOR-4-LOWRSRC_PERCENT: F1: cpp_ha: QFP 0 IRAM resource low - 98 percent depleted
```

대부분의 메모리를 사용하는 프로세스를 확인하려면 **show platform hardware qfp active infra statistics user** 명령을 입력합니다.

```
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
```