

# A memória do 1000 Series Router ASR pesquisa defeitos o guia

## Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Vista geral da disposição de memória ASR](#)

[Alocação de memória sob o pool do lsmpl io](#)

[Utilização de memória](#)

[Verifique a utilização de memória em IOS-XE](#)

[Verifique a utilização de memória em IOSd](#)

[Verifique a utilização de TCAM em um ASR1K](#)

[Verifique a utilização de memória em QFP](#)

## Introdução

Este documento descreve como verificar a memória de sistema e pesquisar defeitos problemas relacionados da memória no Roteadores de serviços de agregação Cisco ASR série 1000 (ASR1K).

## Pré-requisitos

### Requisitos

Cisco recomenda que você tem o conhecimento básico destes assuntos:

- Software Cisco IOS XE
- ASR CLI

**Note:** Você pôde precisar uma licença especial a fim entrar ao shell de Linux no 1001 Series Router ASR.

### [Componentes Utilizados](#)

As informações neste documento são baseadas nestas versões de software e hardware:

- Todas as Plataformas ASR1K
- Todas as liberações do Software Cisco IOS XE que apoiam a plataforma ASR1K

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

## Vista geral da disposição de memória ASR

Com as a maioria das Plataformas do roteador baseado em software, a maioria dos processos do software interno é executada dentro da memória do <sup>®</sup> do Cisco IOS. A plataforma ASR1K introduz uma arquitetura de software distribuída que mova muitas responsabilidades do operating system (OS) fora do processo de IOS. Os IO nesta arquitetura, que era previamente responsável para quase todas as operações internas, são executado agora como um de muitos processos de Linux. Isto permite que outros processos de Linux compartilhem da responsabilidade para o funcionamento do roteador.

O ASR1K executa IOS-XE, não os IO tradicionais. Em IOS-XE, um componente de Linux executa o núcleo, e os IO são executado como um demônio, que seja consultado daqui por diante como IOSd (IO-demônio). Isto cria uma exigência que a memória seja rachada entre o kernel (centro) de Linux e o exemplo de IOSd.

A memória que é rachada entre IOSd e o resto do sistema é fixa na partida e não pode ser alterada. Para um sistema 4-GB, IOSd é atribuído aproximadamente 2 GB, e para um sistema 8-GB, o IOSd é atribuído aproximadamente 4 GB (com a Redundância de software desabilitada).

Desde que o ASR1K tem uma arquitetura 64-bit, todo o ponteiro que estiver em cada estrutura de dados no sistema consome o dobro a quantidade de memória quando comparado ao único CPU as Plataformas tradicionais (8 bytes em vez de 4 bytes). O endereçamento de 64 bits permite IO de superar a limitação de memória 2-GB endereçável dos IO, que permite que escale a milhões de rotas.

**Note:** Assegure-se de que você tenha a memória suficiente disponível antes que você ative todos os novos recursos. Cisco recomenda que você tem pelo menos 8 GB DRAM se você recebe a tabela de roteamento inteira do Border Gateway Protocol (BGP) quando a Redundância de software está permitida a fim impedir a exaustão da memória.

## Alocação de memória sob o pool do lsmapi\_io

O conjunto de memória da relação do pontapé da memória compartilhada de Linux (LSMPI) é usado a fim transferir pacotes do Forwarding Processor ao processador de rotas. Este conjunto de memória é cinzelado na inicialização de roteador em bufferes preallocated, ao contrário do conjunto de processador, onde IOS-XE atribui blocos de memória dinamicamente. Na plataforma ASR1K, o pool do lsmapi\_io tem pouco do do â da memória livre o do do â de geralmente menos de 1000 bytes que é normal. Cisco recomenda que você desabilita a monitoração do pool LSMPI pelos aplicativos de gerenciamento de rede a fim evitar alarmes falsos.

```
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
```

Se há alguma edição no trajeto LSMPI, o contador da falha do xmit do dispositivo parece incrementar neste comando output (alguma saída omitida):

```
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>
```

## Utilização de memória

O ASR1K compreende estes elementos funcionais em seu sistema:

- Route processor (RP) do 1000 Series ASR
- O 1000 Series ASR encaixou o processador de serviços (o ESP)
- Processador de interface dos TERMAS do 1000 Series ASR (SORVO)

Como tal, é exigido para monitorar a utilização de memória por cada um destes processadores em um ambiente de produção.

Os processadores de controle executam o Software Cisco IOS XE que consiste em um núcleo Linux-baseado e em um grupo comum de programas de utilitário do OS-nível, que inclua o Cisco IOS que executa como um processo do usuário no cartão RP.

## Verifique a utilização de memória em IOS-XE

Incorpore o comando do **resumo do Control Processor do estado do software de plataforma da mostra** a fim monitorar a utilização de memória no RP, no ESP, e no SORVO. O estado de sistema deve ser idêntico, com respeito aos aspectos tais como a configuração e o tráfego da característica, quando você comparar a utilização de memória.

```
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%)
```

**Note:** A memória comprometida é uma avaliação de quanto RAM você precisa a fim garantir que o sistema é nunca fora da memória (OOM) para esta carga de trabalho. Normalmente, a memória dos overcommits do núcleo. Por exemplo, quando você executa um malloc 1-GB, nada acontece realmente. Você recebe somente a memória-em-procura verdadeira quando você começa a usar essa memória atribuída, e somente tanto quanto você se usa.

Cada processador alistado na saída precedente pôde relatar o estado como **saudável**, o **aviso**, ou o **crítico**, que é dependente da quantidade de memória livre. Se alguns dos processadores indicam o estado como o **aviso** ou **crítico**, incorpore o comando do **process<slot> do software de plataforma do monitor** a fim identificar o contribuinte superior.

```
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>
```

Você pôde ser alertado ajustar o terminal-tipo antes que você possa executar o comando do **processo do software de plataforma do monitor**:

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

O tipo de terminal é ajustado à **rede** à revelia. A fim ajustar o tipo de terminal apropriado, inscreva o **comando terminal terminal-type**:

## Terminal-tipo VT100 ASR1K#TERMINAL

Uma vez que o tipo de terminal correto é configurado, você pode incorporar o comando do **processo do software de plataforma do monitor** (alguma saída omitida):

```
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>
```

**Note:** A fim classificar a saída no ordem decrescente de utilização de memória, pressione a **SHIFT + o M**.

## Verifique a utilização de memória em IOSd

Se você observa que o processo do **linux\_iosd-imag** guarda raramente uma grande quantidade de memória na saída do **comando active rp** do processo do software de plataforma do monitor, centre seus empenhos no Troubleshooting sobre o exemplo de IOSd. É provável que um processo específico na linha de IOSd não livra acima a memória. Pesquise defeitos problemas relacionados da memória no exemplo de IOSd a mesma maneira que você pesquisa defeitos todas as Plataformas com base no software da transmissão, tais como Cisco 2800, 3800, ou 3900 Series.

```
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>
```

Incorpore o comando **classificado show process memory** a fim identificar o processo do problema:

```
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
```

**Note:** Abra um caso de TAC se você exige o auxílio a fim pesquisar defeitos ou identificar se a utilização de memória é legítima.

## Verifique a utilização de TCAM em um ASR1K

A Classificação de tráfego é uma das maiores de funções básicas encontradas nos Roteadores e nos Switches. Muitos aplicativos e características exigem que os dispositivos de infraestrutura proporcionem estes Serviços diferenciados para os usuários diferentes baseados em exigências de qualidade. O processo da Classificação de tráfego deve ser rápido, de modo que a taxa de transferência do dispositivo não seja degradada extremamente. A plataforma ASR1K usa a 4ª geração do Ternary Content Addressable Memory (TCAM4) por esse motivo.

A fim determinar o número total de pilhas TCAM disponíveis na plataforma, e o número de entradas livres que permanecem, incorpore este comando:

```
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
```

**Note:** Cisco recomenda que você verifique sempre o estado do ponto inicial antes que você faça todas as mudanças às políticas das listas de acesso ou do Qualidade de Serviço (QoS), de modo que o TCAM tenha as suficientes pilhas livres disponíveis a fim programar as entradas.

Se o Forwarding Processor é executado criticamente baixo em pilhas livres TCAM, o ESP pôde gerar os logs similares àqueles mostrados abaixo e pôde causar um crash. Se não há nenhuma Redundância, este conduz ao rompimento raffic.

```
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
```

## Verifique a utilização de memória em QFP

Além do que a memória física, há igualmente uma memória anexada ao processador do fluxo do quantum (QFP) ASIC que é usado a fim enviar estruturas de dados, que inclui dados tais como o banco de informação de encaminhamento (FIB) e as políticas de QoS. A quantidade de DRAM

disponível para o QFP ASIC é fixa, com escalas do 256 MB, do 512 MB e do 1 GB, dependente do módulo ESP.

Inscreva o **comando statistics ativo do exmem da infraestrutura do qfp do hardware da plataforma da mostra** a fim determinar a utilização de memória do **exmem**. A soma da memória para IRAM e DRAM que é usado dá a memória total QFP que está no uso.

```
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
```

O IRAM é a memória da instrução para o software QFP. Caso o DRAM for esgotado, IRAM disponível pode ser usado. Se o IRAM é executado criticamente baixo na memória, você pôde ver este Mensagem de Erro:

```
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
```

A fim determinar o processo que consome a maioria da memória, inscreva o **comando user das estatísticas do exmem do active do qfp do hardware da plataforma da mostra infra**:

```
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
```