

ASR 1000系列路由器内存故障排除指南

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[ASR内存布局概述](#)

[lsmpi io池下的内存分配](#)

[内存使用情况](#)

[验证IOS-XE上的内存使用情况](#)

[检验IOSd上的内存使用情况](#)

[验证ASR1K上的TCAM利用率](#)

[验证QFP上的内存利用率](#)

简介

本文档介绍如何检查系统内存并解决Cisco ASR 1000系列聚合服务路由器(ASR1K)上与内存相关的问题。

先决条件

要求

Cisco 建议您具有以下主题的基础知识：

- Cisco IOS-XE软件
- ASR CLI

注意：您可能需要特殊许可证才能登录ASR 1001系列路由器上的Linux外壳。

使用的组件

本文档中的信息基于以下软件和硬件版本：

- 所有ASR1K平台
- 支持ASR1K平台的所有Cisco IOS-XE软件版本

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原

始 (默认) 配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

ASR内存布局概述

在大多数基于软件的路由器平台中，大多数内部软件进程都在Cisco IOS®内存中运行。ASR1K平台引入了分布式软件架构，可将许多操作系统(OS)职责从IOS流程中移出。此架构中的IOS以前负责几乎所有内部操作，现在作为许多Linux进程之一运行。这样，其他Linux进程就可以分担路由器操作的责任。

ASR1K运行IOS-XE，而不是传统IOS。在IOS-XE中，Linux组件运行内核，而IOS作为守护程序运行，后者又称IOSd(IOS-Daemon)。这就要求在Linux内核和IOSd实例之间拆分内存。

在IOSd和系统其余部分之间拆分的内存在启动时固定，无法修改。对于4 GB系统，IOSd分配约2 GB，对于8 GB系统，IOSd分配约4 GB (禁用软件冗余)。

由于ASR1K具有64位架构，因此与传统的单CPU平台 (8字节而非4字节) 相比，系统中每个数据结构中的任何指针消耗的内存量都是传统单CPU平台的两倍。64位编址使IOS能够克服IOS的2 GB可寻址内存限制，从而可扩展到数百万条路由。

注意：在激活任何新功能之前，请确保您有足够的可用内存。如果在启用软件冗余以防止内存耗尽时收到整个边界网关协议(BGP)路由表，思科建议您至少拥有8 GB DRAM。

lsmpi_io池下的内存分配

使用Linux共享内存传输接口(LSMPI)内存池，以便将数据包从转发处理器传输到路由处理器。此内存池在路由器初始化时被划分为预先分配的缓冲区，而不是IOS-XE动态分配内存块的处理池。在ASR1K平台上，lsmpi_io池几乎没有可用内存 — 通常小于1000字节(正常)。思科建议按网络管理应用禁用对LSMPI池的监控，从而避免错误警报。

```
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

如果LSMPI路径中有任何问题，则此命令输出中**Device xmit fail**计数器显示为递增 (省略了一些输出)：

```
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>

```

内存使用情况

ASR1K在其系统中包括以下功能元素：

- ASR 1000 系列路由处理器 (RP)
- ASR 1000 系列嵌入式服务处理器 (ESP)
- ASR 1000 系列 SPA 接口处理器 (SIP)

因此，需要监控生产环境中每个处理器的内存利用率。

控制处理器运行Cisco IOS-XE软件，该软件由基于Linux的内核和一组常见的OS级实用程序组成，这些实用程序包括作为用户进程在RP卡上运行的Cisco IOS。

验证IOS-XE上的内存使用情况

输入**show platform software status control-processor brief**命令以监控RP、ESP和SIP上的内存使用情况。在比较内存使用情况时，系统状态必须相同，例如功能配置和流量。

```

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%)

```

注意：承诺内存是您估计需要多少RAM，以确保系统不会因此工作负荷而内存不足(OOM)。

通常，内核会超量提交内存。例如，当运行1-GB的malloc时，实际上不会发生任何情况。只有开始使用分配的内存时，才会收到真正的按需内存，而且只有使用的内存数。

上一个输出中列出的每个处理器可能报告状态为**Healthy**、**Warning**或**Critical**，这取决于可用内存的大小。如果任何处理器显示状态为**警告**或**严重**，请输入**monitor platform software process<slot>**命令以确定最主要的参与者。

```
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>
```

在执行监控平台软件进程命令之前，可能会提示您设置**terminal-type**:

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

默认情况下，终端类型设置为网络。要设置适当的终端类型，请输入**terminal terminal-type**命令：

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

配置正确的终端类型后，可以输入**monitor platform software process**命令(省略部分输出):

```
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_iods-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>
```

注意：要按内存使用降序对输出进行排序，请按**Shift + M**。

检验IOSd上的内存使用情况

如果您注意到linux_iosd-imag 进程在监控平台软件进程rp active命令输出中存储了异常大的内存，请将故障排除工作集中在IOSd实例上。IOSd线程中的特定进程可能不会释放内存。IOSd实例中与内存相关的问题的故障排除方法与任何基于软件的转发平台（如Cisco 2800、3800或3900系列）的故障排除方法相同。

```
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>
```

输入show process memory sorted命令以确定问题进程：

```
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
```

注意：如果您需要帮助以排除故障或确定内存使用是否合法，请提交TAC案例。

验证ASR1K上的TCAM利用率

流量分类是路由器和交换机中最基本的功能之一。许多应用和功能要求基础设施设备根据质量要求为不同用户提供这些差异化服务。流量分类过程应该很快，这样设备的吞吐量不会大大降低。ASR1K平台使用第4^代三态内容可寻址存储器(TCAM4)来实现此目的。

要确定平台上可用的TCAM信元总数和剩余的可用条目数，请输入以下命令：

```
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
```

注意：思科建议在对访问列表或服务质量(QoS)策略进行任何更改之前始终检查阈值状态，以便TCAM有足够的可用信元来对条目进行编程。

如果转发处理器在空闲TCAM信元上运行极低，ESP可能会生成类似于下图所示的日志，并可能崩溃。如果没有冗余，则会导致流量中断。

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

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

验证QFP上的内存利用率

除物理内存外，还有附加到量子流处理器(QFP)ASIC的内存，该内存用于转发数据结构，包括转发信息库(FIB)和QoS策略等数据。QFP ASIC的可用DRAM量是固定的，范围为256 MB、512 MB和1 GB，具体取决于ESP模块。

输入show platform hardware qfp active infrastructure exmem statistics命令，以确定exmem内存的使用情况。使用的IRAM和DRAM的内存总和可提供正在使用的总QFP内存。

```
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是QFP软件的指令存储器。如果DRAM耗尽，可以使用可用的IRAM。如果IRAM内存严重不足，您可能会看到以下错误消息：

```
%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
```

要确定消耗大部分内存的进程，请输入show platform hardware qfp active infra exmem statistics user命令：

```
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
```