

ASR 1000系列路由器内存排除故障指南

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[ASR内存布局概述](#)

[在lsmpi_io池下的存储器分配](#)

[内存使用情况](#)

[验证在IOS-XE的内存使用](#)

[验证在IOSd的内存使用](#)

[验证在ASR1K的TCAM利用率](#)

[验证在QFP的存储器利用率](#)

简介

本文描述如何检查系统内存和排除故障在Cisco ASR 1000系列汇聚服务路由器(ASR1K)的内存相关问题。

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

Note:您也许需要一个特殊许可证为了登陆到ASR 1001系列路由器的Linux shell。

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

使用大多基于软件的路由器平台,内部软件进程的多数在Cisco IOS内存内运行。ASR1K平台传入移动许多操作系统(OS)责任在IOS进程外面的分布式软件体系结构。IOS在此体系结构方面,对几乎所有以前负责内部操作,当前运行作为许多Linux进程之一。这允许其他Linux进程共同负担对路由器的操作的责任。

ASR1K运行IOS-XE,不是传统IOS。在IOS-XE中,Linux组件运行内核,并且IOS运行作为守护程序,此后参考作为IOSd(IOS守护程序)。这创建内存拆分在Linux内核和IOSd实例之间的要求。

拆分在IOSd和系统的其余之间的内存修复在启动,并且不可能被修改。对于4-GB系统,分配IOSd大约2 GB,并且对于8-GB系统,分配IOSd大约4 GB(当软件冗余禁用)。

因为ASR1K有64位体系结构,在系统的每个数据结构的所有指示器消耗双内存数量,当与传统单个CPU比较平台(而不是4个字节的8个字节)。64位寻址使IOS解决2-GB可寻址的内存限制关于IOS,允许它扩展到数百万路由。

Note:保证您有足够的内存联机,在您激活所有新特性前。思科建议您有至少8个GB DRAM,如果接收整个边界网关协议(BGP)路由表,当软件冗余启用为了防止内存耗尽时。

在lsmpi_io池下的存储器分配

Linux共享内存平底船接口(LSMPI)内存池用于为了传输从转发处理器的数据包到路由处理器。此内存池被雕刻在路由器初始化到被预先分配的缓冲区,与处理器池相对,IOS-XE动态地分配存储器块。在ASR1K平台上,lsmpi_io池有一点空闲存储器,通常少于1000个字节是正常的。思科建议您由网络管理应用程序禁用LSMPI池的监听为了避免假告警。

```
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

如果有任何问题在LSMPI路径,设备xmit失败计数器在此命令输出(省略的若干输出中看上去增加):

```
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包括在其system:的这些功能元件

- 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)

同样地，它由在生产环境的这些处理器中的每一个要求监控存储器利用率。

控制处理器运行包括一个基于linux的内核和共同的一套OS级别实用工具程序，包括Cisco IOS运行作为在RP卡的一用户进程的Cisco IOS XE软件。

验证在IOS-XE的内存使用

输入显示平台软件状态控制处理器摘要命令为了监控在RP、ESP和SIP的内存使用。而您比较内存使用，系统状态一定是相同的，关于方面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%)

```

Note:已承诺内存是多少RAM的估计您需要为了保证系统从未是在内存(OOM)外面此工作量的

。通常，内核overcommits内存。例如，当您运行1-GB malloc时，什么都确实不发生。您只接收真的内存在需求，当您开始使用该已分配内存时和仅，和您使用一样多。

在上一个输出中列出的每个处理器也许报告状态如**健康**，**亚里桑**或者**关键**，取决于空闲内存容量。如果其中任一个处理器显示状态作为**亚里桑**或**关键**，请输入**监视器平台软件process<slot>**命令为了识别顶部投稿人。

```
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>
```

在您能执行**监视器平台软件进程命令**前，也许提示您设置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**。为了设置适当的终端类型，请输入**terminal terminal-type命令**：

```
ASR1K#TERMINAL terminal-type VT100
```

一旦正确终端类型配置，您能输入**监视器平台软件进程命令**(省略的若干输出)：

```
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>
```

Note:为了排序输出降序按内存使用的顺序，请按班次+ M。

验证在IOSd的内存使用

如果注意linux_iosd-imag进程在监视器平台软件进程RP active命令输出中保持一个非常地很多内存，请集中您的故障排除工作于IOSd实例。很可能在IOSd线索的一特定进程不释放内存。排除故障在IOSd实例的内存相关问题同一个方式您排除故障所有基于软件的转发平台，例如Cisco 2800，3800，或者3900系列。

```
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排序的命令为了识别问题进程：

```
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
```

Note:请开TAC案例，如果要求协助为了排除故障或识别，如果内存使用合法。

验证在ASR1K的TCAM利用率

数据流分类是在路由器和交换机找到的多数基本功能之一。许多应用程序和功能要求基础设备为根据质量要求的不同的用户提供这些差异化服务。数据流分类进程应该快速，因此设备的吞吐量不非常地降低。ASR1K平台为此使用三个按内容寻址的存储器(TCAM4)的第4生成。

为了确定TCAM信元的总数可用在平台和依然存在自由输入的数量，请输入此命令：

```
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
```

Note:思科建议您总是检查阈值状态，在您做对访问列表或服务质量(QoS)策略前的所有变动，因此TCAM有满足的自由信元联机为了编程条目。

如果转发处理器运行精密地低在自由的TCAM信元，ESP也许生成日志类似于如下所示的那些并且也许失败。如果没有冗余，这导致raffic中断。

```
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
```

验证在QFP的存储器利用率

除物理存储器之外，也有内存附加对使用为了转发数据结构，包括数据例如转发信息库(FIB)和QoS策略的Quantum流处理器(QFP) ASIC。相当数量QFP的ASIC DRAM联机修复，与范围256 MB、512 MB和1 GB，从属在ESP模块。

输入显示平台硬件qfp statistics命令活动基础设施的exmem为了确定exmem内存使用。内存的总和使用的IRAM和DRAM的给是在使用中的总计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软件的说明内存。在DRAM用尽情况下，可以使用联机IRAM。如果IRAM运行精密地低在内存，您也许发现此错误消息：

```
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
```

为了确定消耗大多数内存的进程，请输入user命令显示平台硬件qfp激活在下exmem的统计信息：

```
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