

了解9800 WLC上的内存泄漏问题

目录

[简介](#)

[内存泄漏](#)

[系统日志](#)

[如何识别9800 WLC遇到内存泄漏问题](#)

[IOS进程中内存泄漏故障排除](#)

[来自WLC的基本日志](#)

[对于处理器内存池](#)

[对于IO内存池](#)

[Polaris/平台级内存泄漏故障排除](#)

[建议](#)

简介

本文档介绍Cisco Catalyst 9800无线LAN控制器(WLC)中的内存泄漏。

内存泄漏

当程序或进程分配内存供临时使用时，并且当不再需要时，未正确取消分配内存时，从操作系统的角度来看，该内存仍处于“使用中”。由于进程继续运行并且反复无法取消分配内存，该进程使用的内存总量会增加，而且可用于其他进程和系统功能的内存会减少。内存泄漏通常是由软件错误或系统固件或在其上运行的应用程序中的问题引起的。

在Cisco Catalyst 9800 WLC中，内存泄漏可能表现为以下情况：

- 性能降低：由于内存越来越稀缺，WLC可能会减速，从而导致管理功能的响应时间变慢或连接到网络的客户端设备的性能降低。
- 系统不稳定：关键进程可能会开始失败，可能导致客户端连接断开、无法管理WLC或其他不稳定行为。
- 系统崩溃：在严重情况下，WLC可能会崩溃和重新启动，尤其是当内存不足以进行基本操作时。

注意：9800 WLC可能会突然重新启动/崩溃，以回收泄漏的内存并自行恢复。由于内存泄漏是错误行为，因此即使重新启动后也会发生泄漏，除非已禁用导致泄漏的配置/功能。

系统日志

%PLATFORM-4-ELEMENT_WARNING : R0/0 : smand : RP/0使用的内存值91%超过警告级别88%

此消息将打印前3个内存使用进程的名称以及tracekey、callsite ID和diff调用：

%PLATFORM-4-ELEMENT_WARNING : 机箱1 R0/0 : smand : 1/RP/0 : 已用内存值91%超过警告级别88%。排名靠前的内存分配程序为：进程：sessmgrd_rp_0。Tracekey：1#258b8858a63c7998252e96352473c9c6 Callsite ID：11B8F825A8768000 (diff_call：20941)。过程：fman_fp_image_fp_0。Tracekey：1#36b34d8e636a89f6397a3b12acab9706 Callsite ID：1944E78DF68EC002 (diff_call：19887)。进程：linux_iosd-imag_rp_0。Tracekey：1#8ec74901dc8e23a44e060e69d5820ece Callsite ID：E2AA338E11594003 (diff_call：13404)。

如何识别9800 WLC遇到内存泄漏问题

必须及时解决内存泄漏问题，因为它们可能会影响WLC提供的网络服务的稳定性和可靠性。要诊断WLC上的内存泄漏，可以在CLI上使用各种命令来监控一段时间的内存使用情况。他们可能会寻找在不释放内存的情况下使用不断增加的内存的进程，或者寻找表明内存未按预期回收的模式。

检查已完全分配给平台的内存量。

```
9800WLC#show version | in memory
cisco C9800-L-F-K9 (KATAR) processor (revision KATAR) with 1634914K/6147K bytes of memory.

32768K bytes of non-volatile configuration memory.
16777216K bytes of physical memory.
```

!! Determines Total platform memory available, Here it is 16GB

检查为每个池分配了多少内存。

```
9800WLC#show processes memory
Processor Pool Total: 1674013452 Used: 823578520 Free: 850434932
reserve P Pool Total: 102404 Used: 88 Free: 102316
lsmapi_io Pool Total: 6295128 Used: 6294296 Free: 832
```

检查资源利用率，包括内存使用情况。如果它超过“警告”或“严重”级别，则表明可能存在内存泄漏。

```
9800WLC#show platform resources
**State Acronym: H - Healthy, W - Warning, C - Critical
Resource Usage Max Warning Critical State
-----
RP0 (ok, active)
Control Processor 21.70% 100% 80% 90% H
DRAM 5444MB (35%) 15467MB 88% 93% H
ESP0(ok, active)
QFP
DRAM 234658KB (12%) 1835008KB 85% 95% H
IRAM 414KB (20%) 2048KB 85% 95% H
CPU Utilization 0.00% 100% 90% 95% H
```

9800 WLC上的内存利用率

监控控制平面资源的整体内存使用情况

```
9800WLC#show platform software status control-processor brief
```

```
Slot Status 1-Min 5-Min 15-Min
1-RP0 Healthy 0.52 0.75 0.80
```

Memory (kB)

```
Slot Status Total Used (Pct) Free (Pct) Committed (Pct)
```

1-RPO Healthy 16327028 4898110(30%) 114218918 (70%) 5387920 (33%)

监控为排名靠前的进程分配的和使用的内存大小。如果内存使用率在可用内存保持固定或过低的情况下继续增加，则在IOSd级别出现内存泄漏的可能性很高。

```
9800WLC#show process memory sorted
Processor Pool Total: 1674013452 Used: 492934952 Free: 1181078500
reserve P Pool Total: 102404 Used: 88 Free: 102316
lsmpi_io Pool Total: 6295128 Used: 6294296 Free: 832

PID TTY Allocated Freed Holding Getbufs Retbufs Process
0 0 737247000 444817776 268572424 0 0 *Init*
736 0 147160744 85216176 43848536 0 0 Stby Cnfg Parse
722 0 34348696 205824 34480984 0 0 SBC main process
4 0 62523104 35323288 23572272 27362640 27360228 RF Slave Main Th
81 0 22061704 91560 21946768 0 0 EWLC IOSD CAPWAP
93 0 70079512 14591040 19359760 0 0 IOSD ipc task
0 0 0 0 6236576 0 0 *MallocLite*
224 0 10665096 619664 6202672 0 0 SNMP MA SA
```

从最高保持进程开始的每进程内存统计信息

对于平台级内存泄漏问题，请监控RSS（驻留集大小）计数器。RSS指示在执行过程中分配给进程的内存量。如果该值迅速增加，则可能表示潜在的内存泄漏。

```
9800WLC#show process memory platform sorted
System memory: 15838752K total, 5409956K used, 10428796K free,
Lowest: 10379012K
Pid Text Data Stack Dynamic RSS Name
-----
4272 409975 1482448 136 468 1482448 linux_iosd-imag
19727 22205 448216 136 1680 448216 ucode_pkt_PPE0
19880 182 373884 136 5772 373884 wncmgrd
20381 991 370916 136 16416 370916 wncd_0
24705 536 334212 136 6928 334212 dbm
21097 342 302808 136 1432 302808 cpp_cp_svr
26601 91 295656 136 19228 295656 pubd
31626 58 274280 136 6744 274280 paed
26889 361 263072 136 368 263072 ndbmand
23222 478 259024 136 11136 259024 repm
24961 57 229112 136 228 229112 cli_agent
```

平台从最高保留进程处理内存使用率

IOS进程中内存泄漏故障排除

在IOS XE中，IOS作为在Linux内核上运行的进程（守护程序）运行，称为IOSd。通常，IOSd分配在可用平台DRAM总量的35%到50%之间。

来自WLC的基本日志

启用时间戳，以具有所有命令的时间参考。

```
9800WLC#term exec prompt timestamp
```

要查看配置和内存相关信息，请执行以下操作：

```
9800WLC#show tech-support wireless
```

```
9800WLC#show tech-support memory
```

收集核心转储文件或系统报告（如果已生成）

通过GUI

导航到 Troubleshooting > Core Dumps and System Report

Troubleshooting > Core Dump and System Report

[Need help on what logs to collect for various scenarios?](#)

Core Dump				
Delete				
	Date & Time	Size (Bytes)	Name	Download
<input type="checkbox"/>	29 Apr 2024 23:56:21	125665	bootflash-2/core/WLC-2_1_RP_0_code_sign_verify_894_20240429-182620-UTC.core.gz	Download
1 - 1 of 1 items				

System Report				
Delete				
	Date & Time	Size (Bytes)	Name	Download
<input type="checkbox"/>	03 Jul 2024 00:38:23	14560784	bootflash/core/WLC-2_1_RP_0-system-report_20240703-003816-IST.tar.gz	Download
<input type="checkbox"/>	25 Jun 2024 23:54:31	16580832	bootflash/core/WLC-2_1_RP_0-system-report_20240625-235418-IST.tar.gz	Download

核心转储和系统报告

通过CLI

```
9800WLC#show bootflash: | in core/system-report
```

```
9800WLC#copy bootflash:system-report/Core_file {tftp: | ftp: | https: ...}
```

对于处理器内存池

从最高保持进程开始，按进程检查内存。

```
9800WLC#show process memory sorted
```

检查相关池的总内存统计信息。它还显示自启动以来最大的可用块和最低的可用内存。

```
9800WLC#show memory Statistics
```

检查分配了大量内存的程序计数器(PC)。

```
9800WLC#show memory allocation-process totals
```

检查泄露的块和块。

```
9800WLC#show memory debug leak chunks
```

!!This is CPU intensive cli and use only if above CLI output is not helping.

对于IO内存池

检查排名靠前的分配器。

```
9800WLC#show memory io allocating-process totals
```

如果顶部分配器是“数据包数据或池管理器”，请检查哪个caller_pc请求了大量的缓冲区

```
9800WLC#show buffers
```

```
9800WLC#show buffers usage
```

如果顶部分配器为 'managed_chunk_process()' 或'Chunk Manager' 进程，则意味着一个或多个区块正在分配大量内存。

```
9800WLC#show chunk summary
```

```
9800WLC#show chunk brief
```

如果进程MallocLite是顶级分配器

```
9800WLC#show memory lite-chunks totals
9800WLC#show memory lite-chunks stats
```

Polaris/平台级内存泄漏故障排除

检查平台上的可用内存资源的内存使用率%。

```
9800WLC#show Platform resources
```

检查整体系统内存快照。

```
9800WLC#show platform software process slot chassis active R0 Monitor | in Mem
```

检查已排序的所有平台进程内存。

```
9800WLC#show process memory platform sorted
9800WLC#show platform software process memory chassis active r0 all sorted
```

检查最近每小时呼叫站点的状态。

```
9800WLC#show process memory platform accounting
```

从前两个CLI输出中选择排名靠前的竞争者，然后启用各个进程的调试。

```
9800WLC#debug platform software memory <process> chassis <1-2/active/standby> R0 alloc callsite stop
9800WLC#debug platform software memory <process> chassis <1-2/active/standby> R0 alloc callsite clear
9800WLC#debug platform software memory <process> chassis <1-2/active/standby> R0 alloc backtrace start <CALL_SITE> depth 10
9800WLC#debug platform software memory <process> chassis <1-2/active/standby> R0 alloc callsite start
!! Running these debugs has no impact to device
```

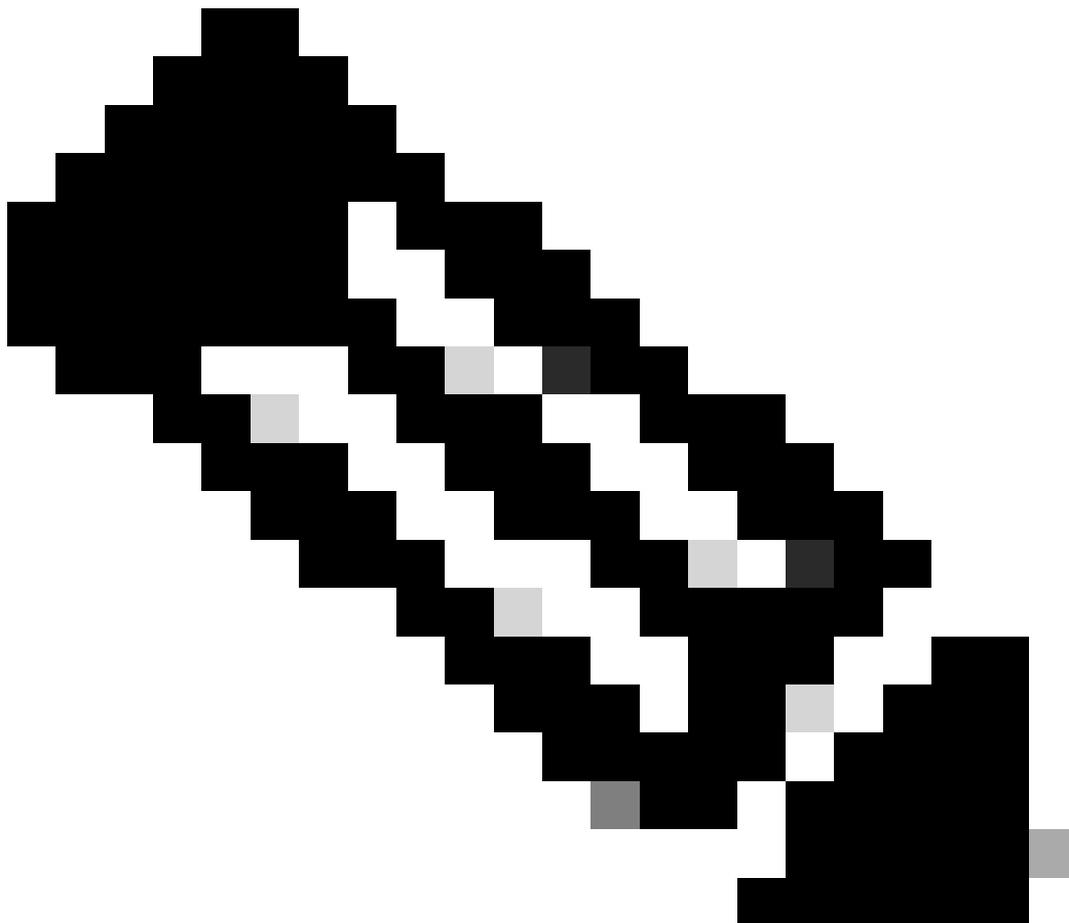
在启动调试后几分钟（15分钟到1小时）收集输出。

```
9800WLC#show platform software memory <process> chassis <1-2/active/standby> R0 alloc backtrace
```

!! Capture this output three times, with a 5-10 minutes interval between each capture, to identify the pattern.

检查call_diff、allocs和frees值以及每个进程的相应回溯。

9800WLC#show platform software memory <process> chassis <1-2/active/standby> R0 alloc callsite brief



注意 : call_diff = allocs - 释放

如果allocs = 释放 , 则无内存泄漏

如果释放= 0，内存泄漏

如果allocs!=释放内存，可能是也可能不是内存泄漏（如果call_diff更多，则表示内存泄漏的可能性很高）

为各个进程捕获数据库内存的数据。

```
9800WLC#show platform software memory <process> chassis <1-2/active/standby>active R0 alloc type data brief
9800WLC#show platform software memory database <process> chassis <1-2/active/standby> chassis active R0 brief
```

检查系统装载信息，以检查临时创建的虚拟文件系统的内存使用情况。

```
9800WLC#show platform software mount
```

建议

有关内存建议和扩展限制，请参阅相关配置指南、产品手册和发行说明，并确保WLC升级到最新推荐版本。

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。